kaspersky

Message Broker

C++ Enterprise Edition

Бычук Александр
Software Architect

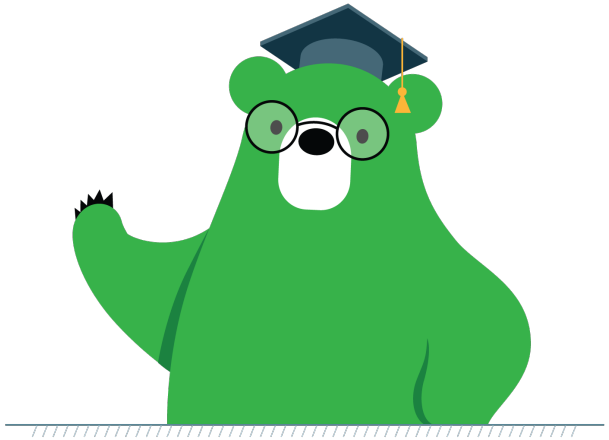
kaspersky

Кто я такой, чтобы говорить про брокеры ?

Печенька за пулл-реквест

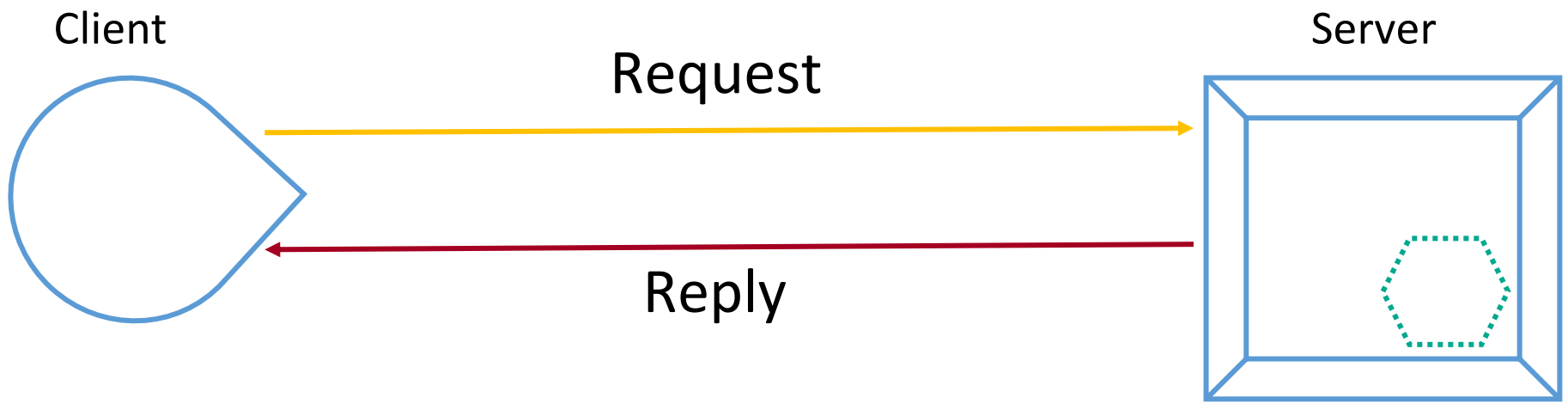
https://github.com/ivk-jsc/broker

Что такое Брокер сообщений ?



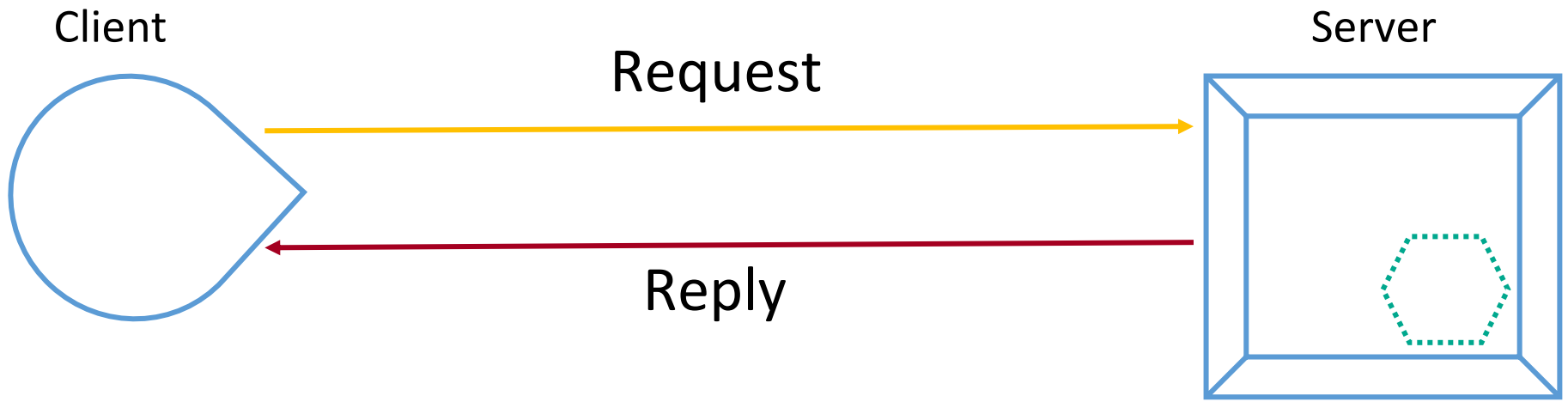
Client-Server

Simple Mode



Client-Server

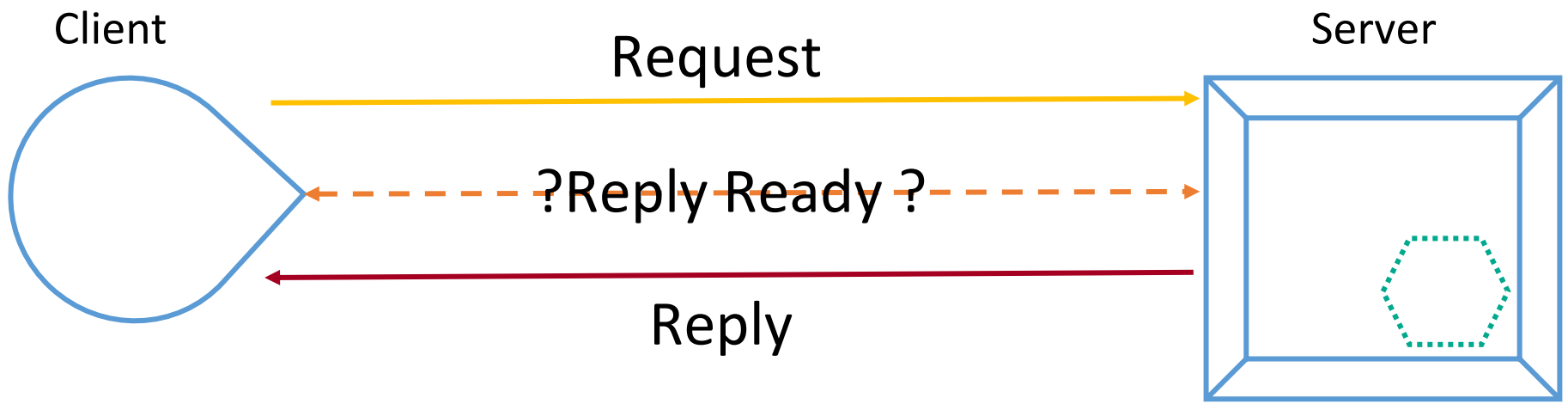
Simple Mode



NOW !

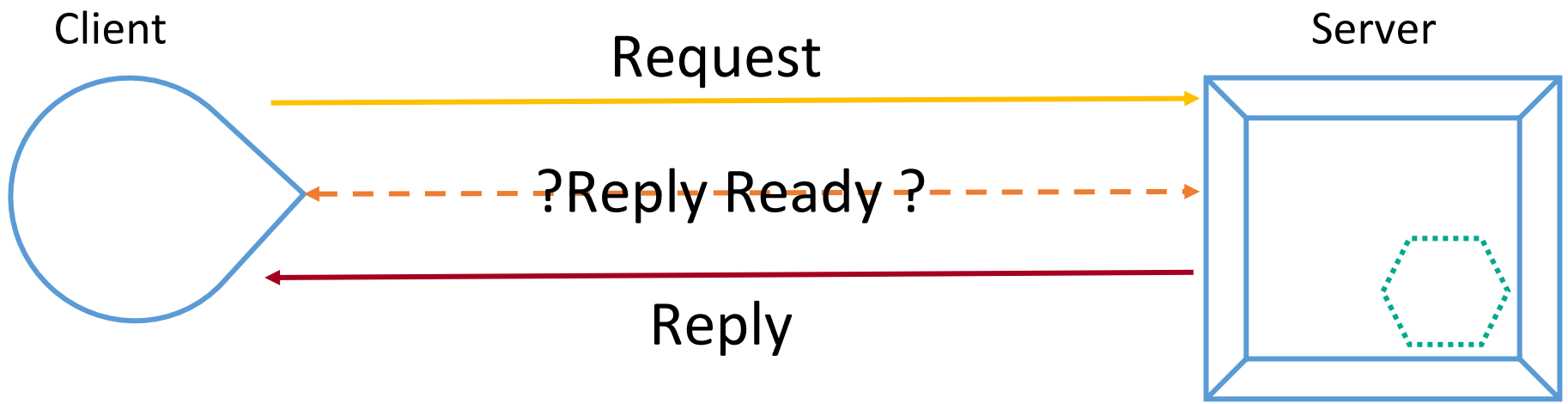
Client-Server

Advanced Mode



Client-Server

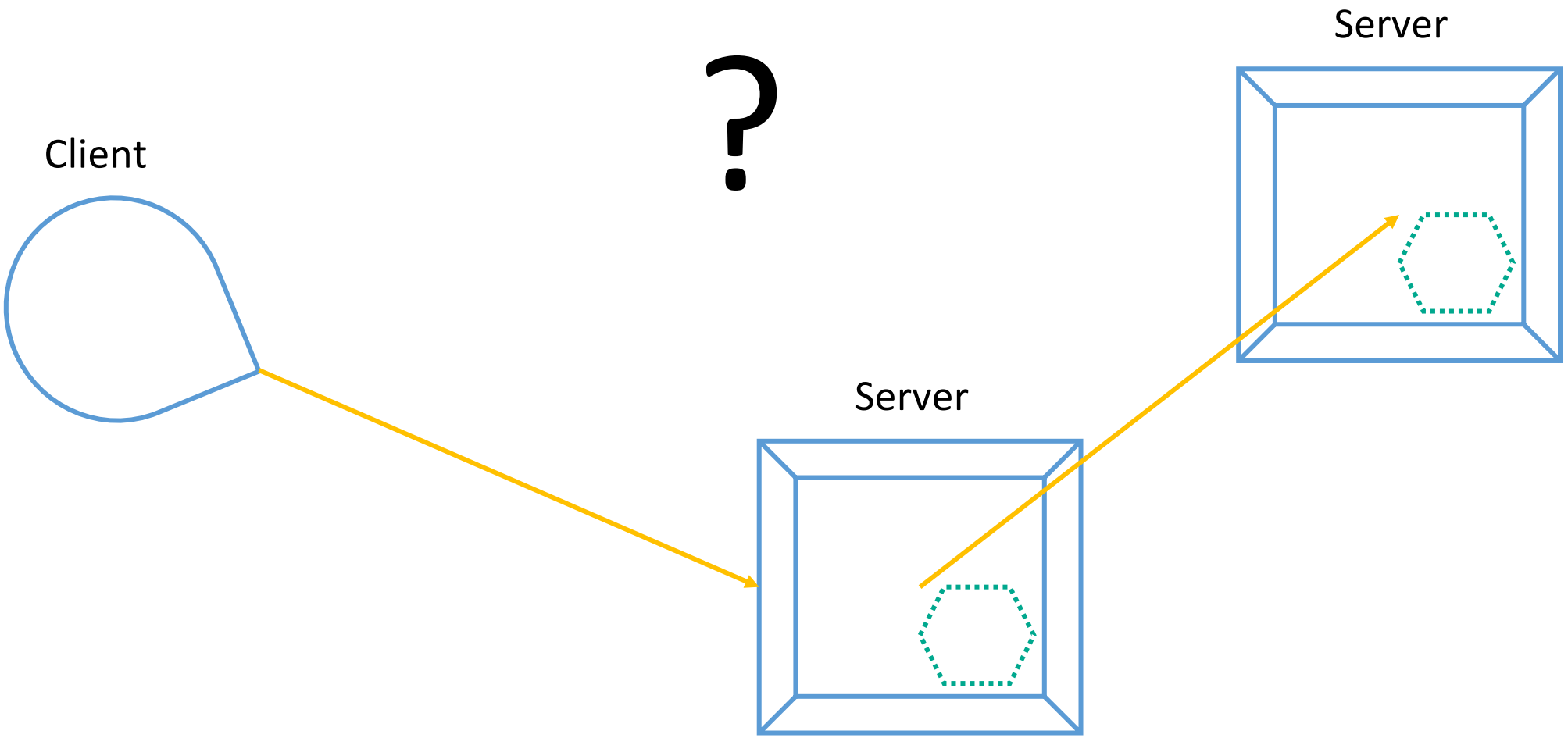
Advanced Mode



Traffic Jam !

Client-Server

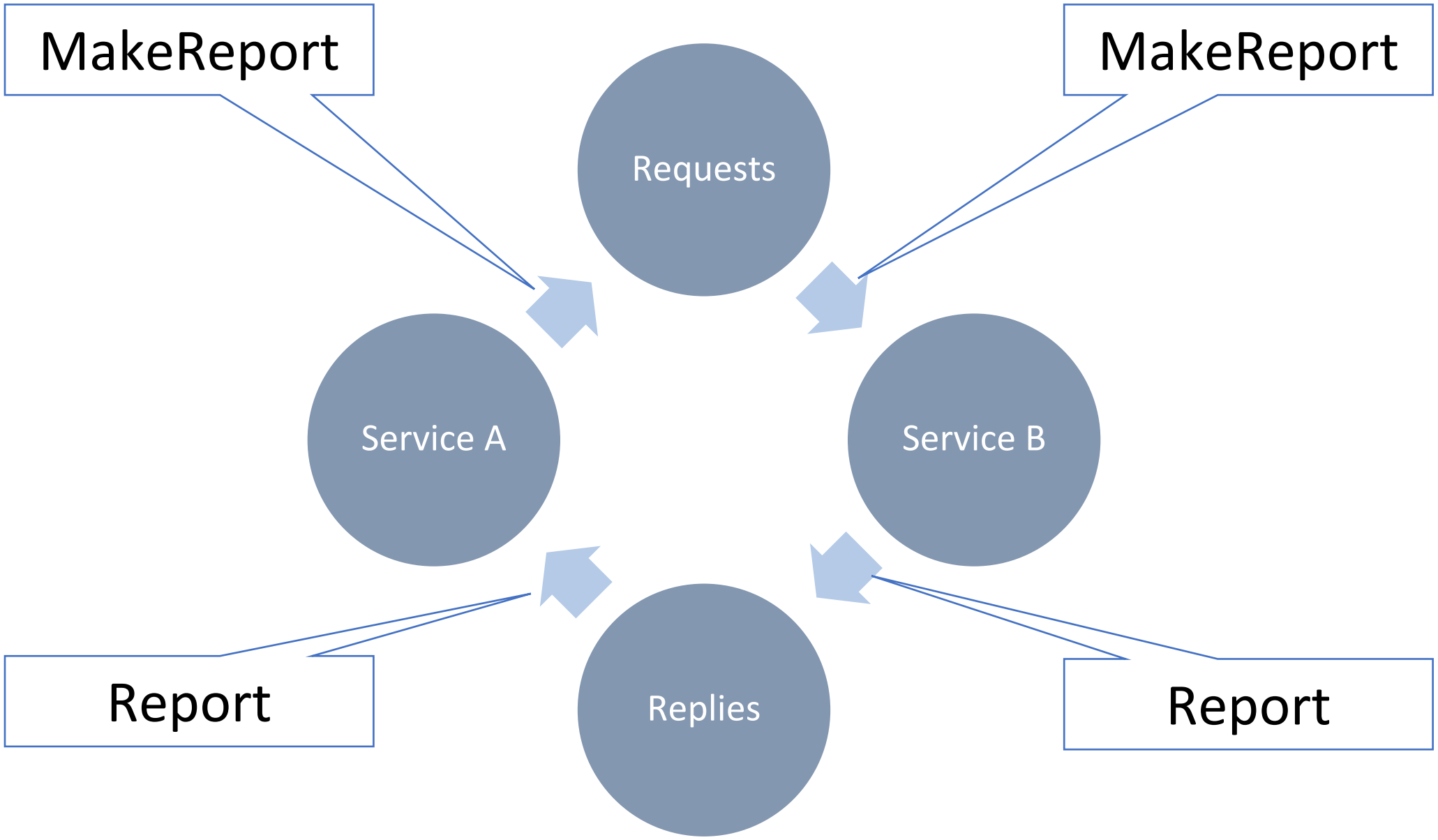
Distributed Mode



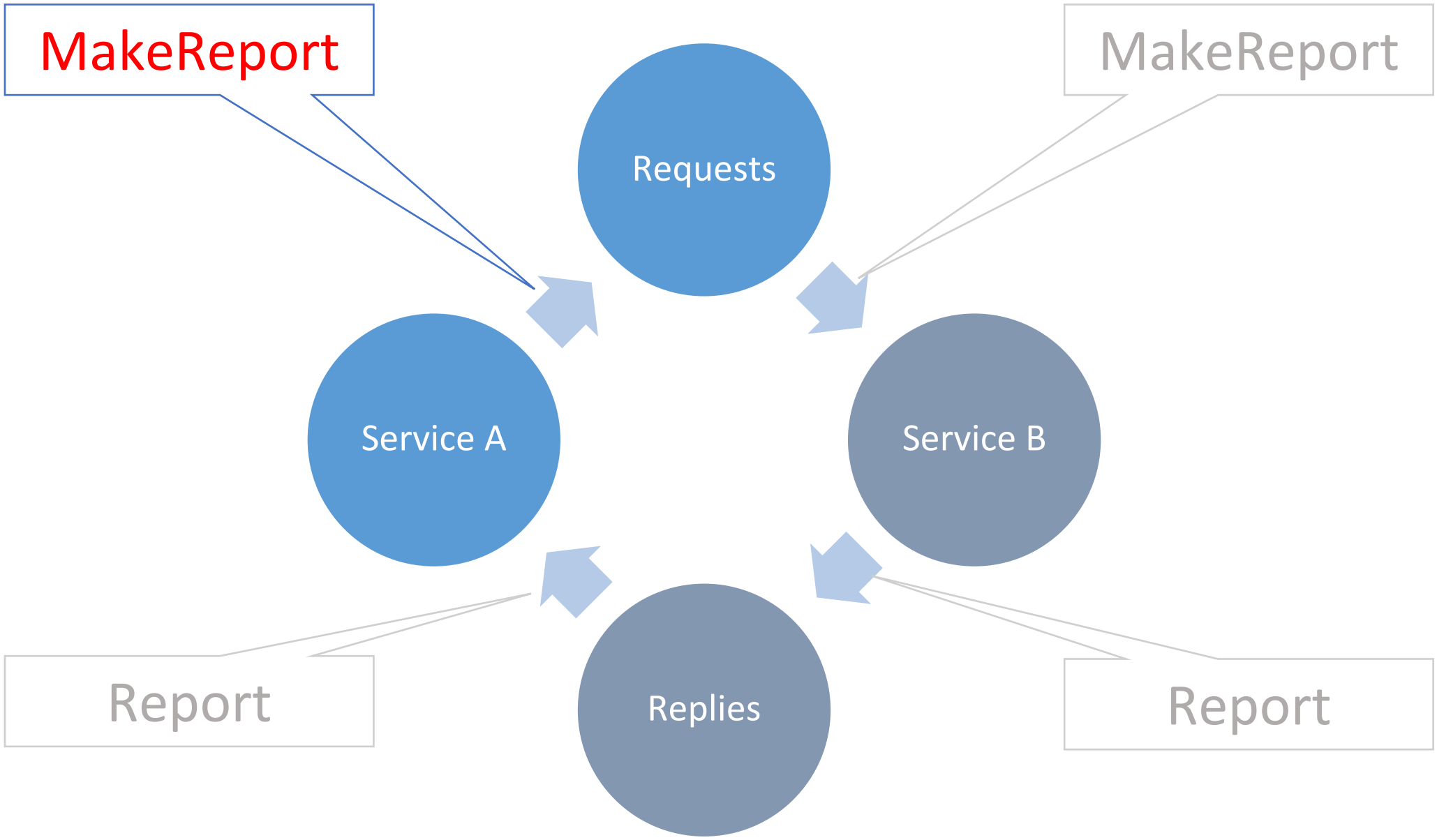
Корпоративные информационные системы

- Асинхронный обмен между сервисами
- Гарантированная доставка сообщений
- Широковещательная рассылка
- Распределённые транзакции

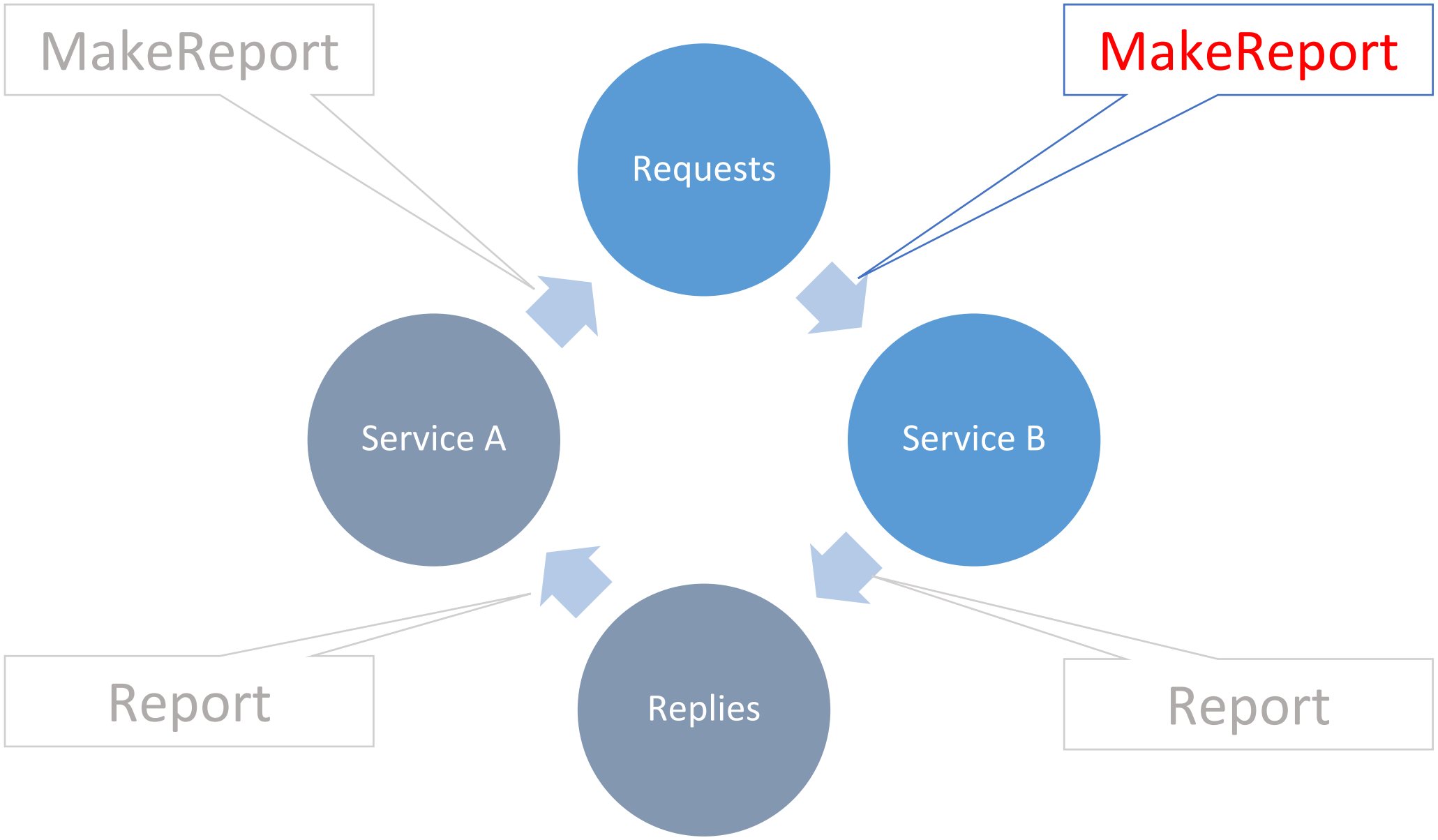
Service to Service



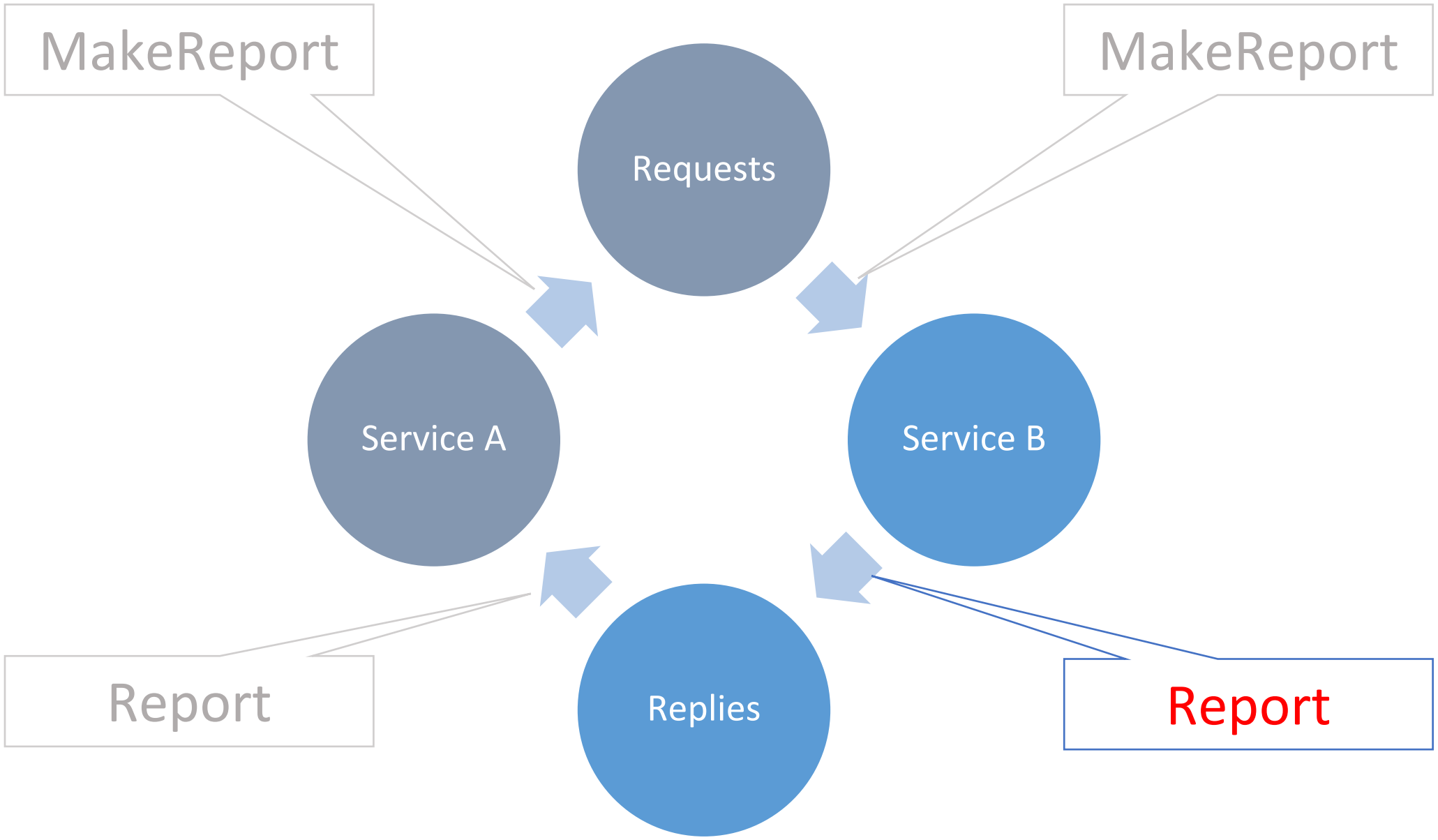
Service to Service



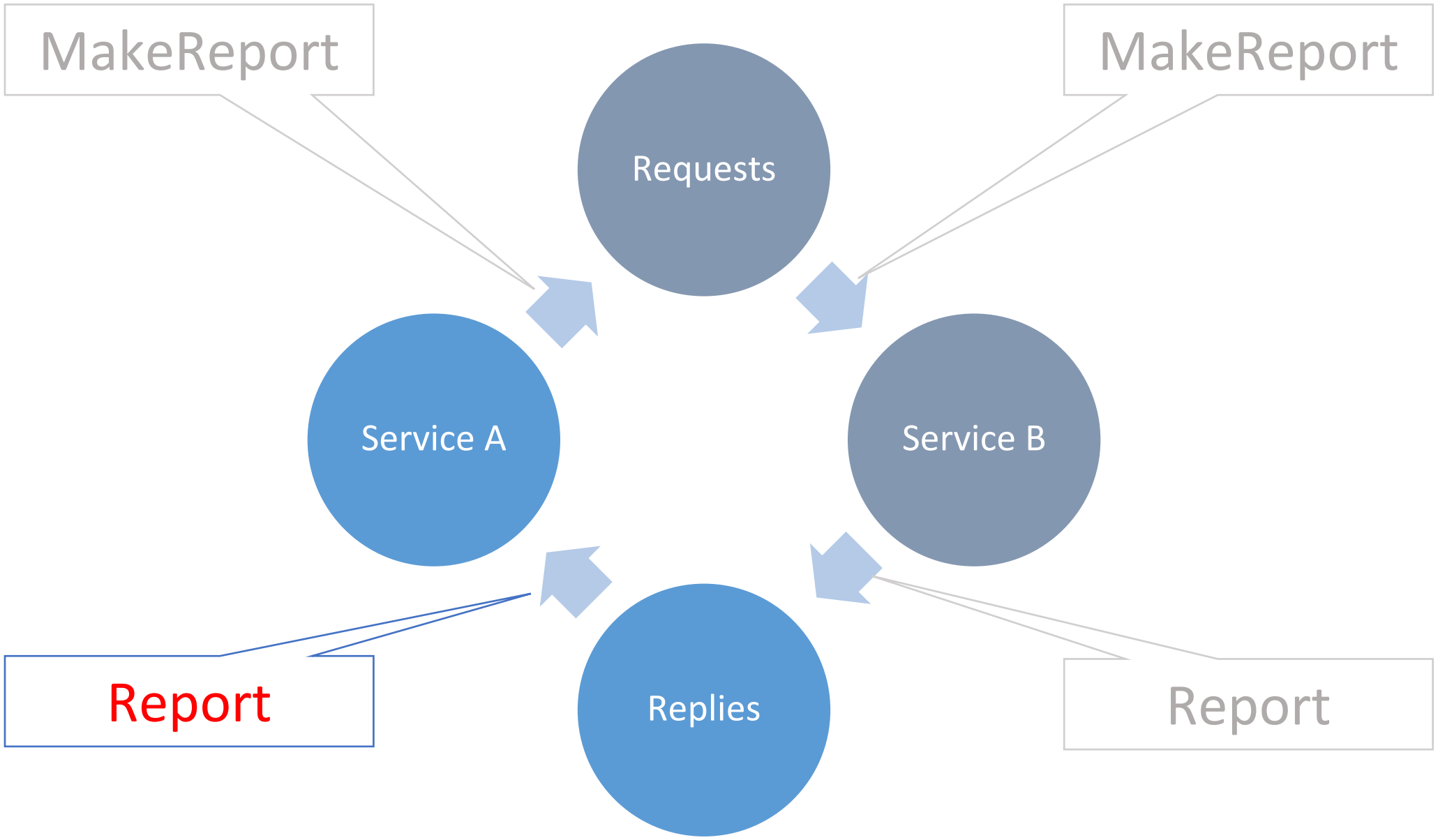
Service to Service



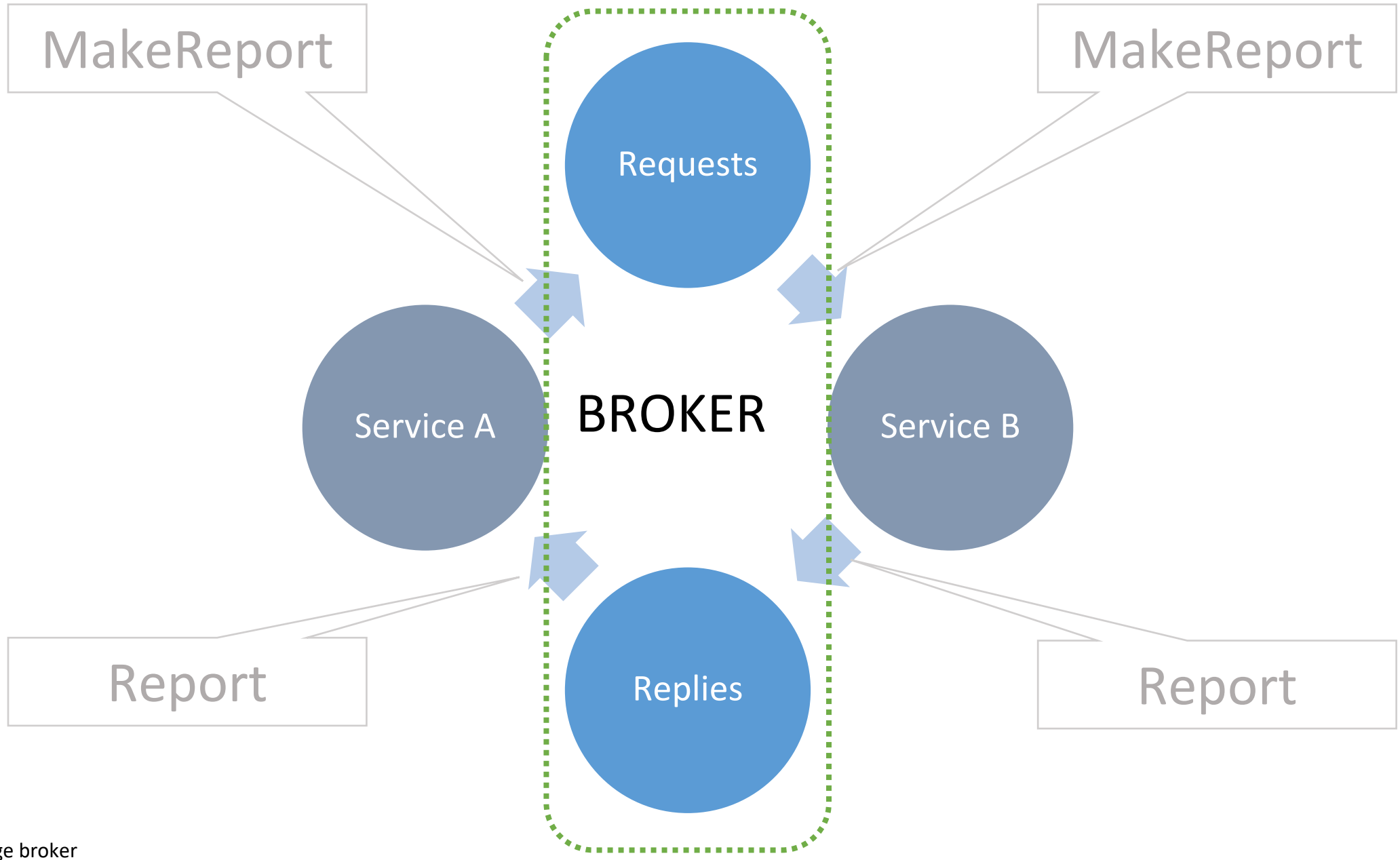
Service to Service



Service to Service

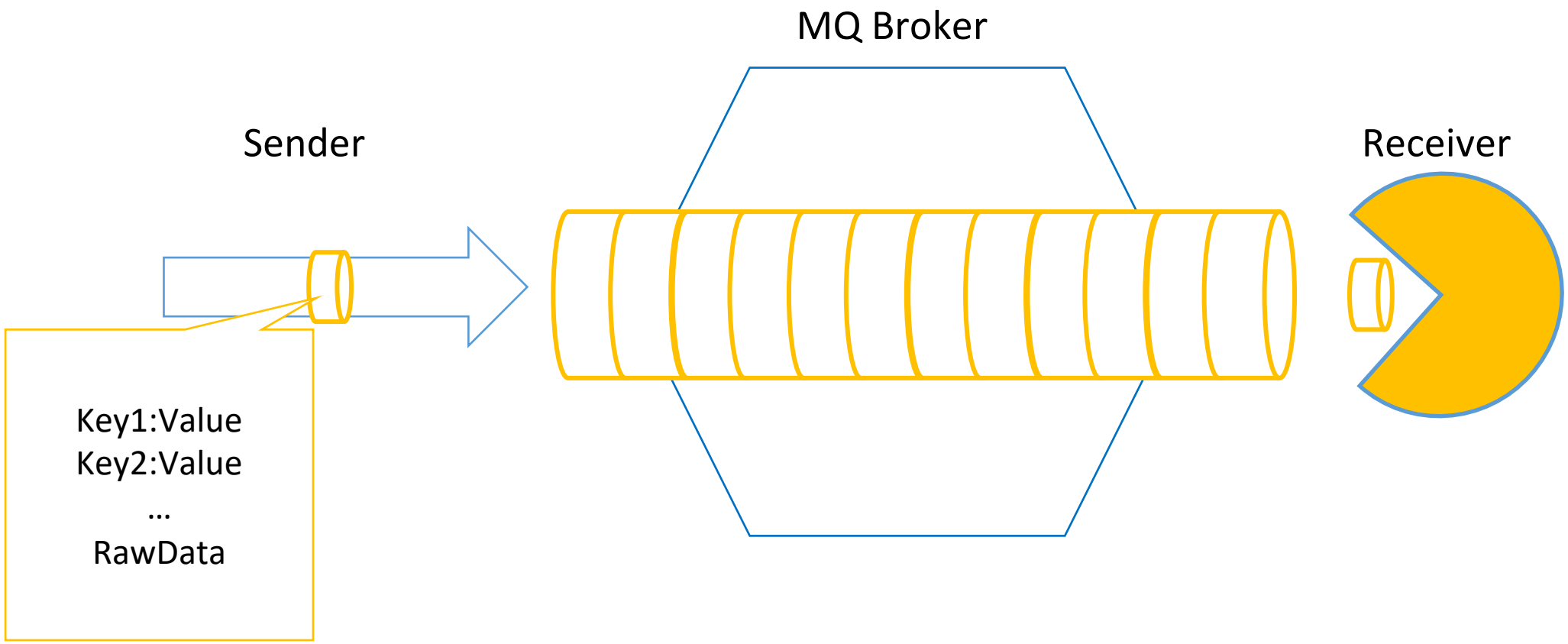


Service to Service

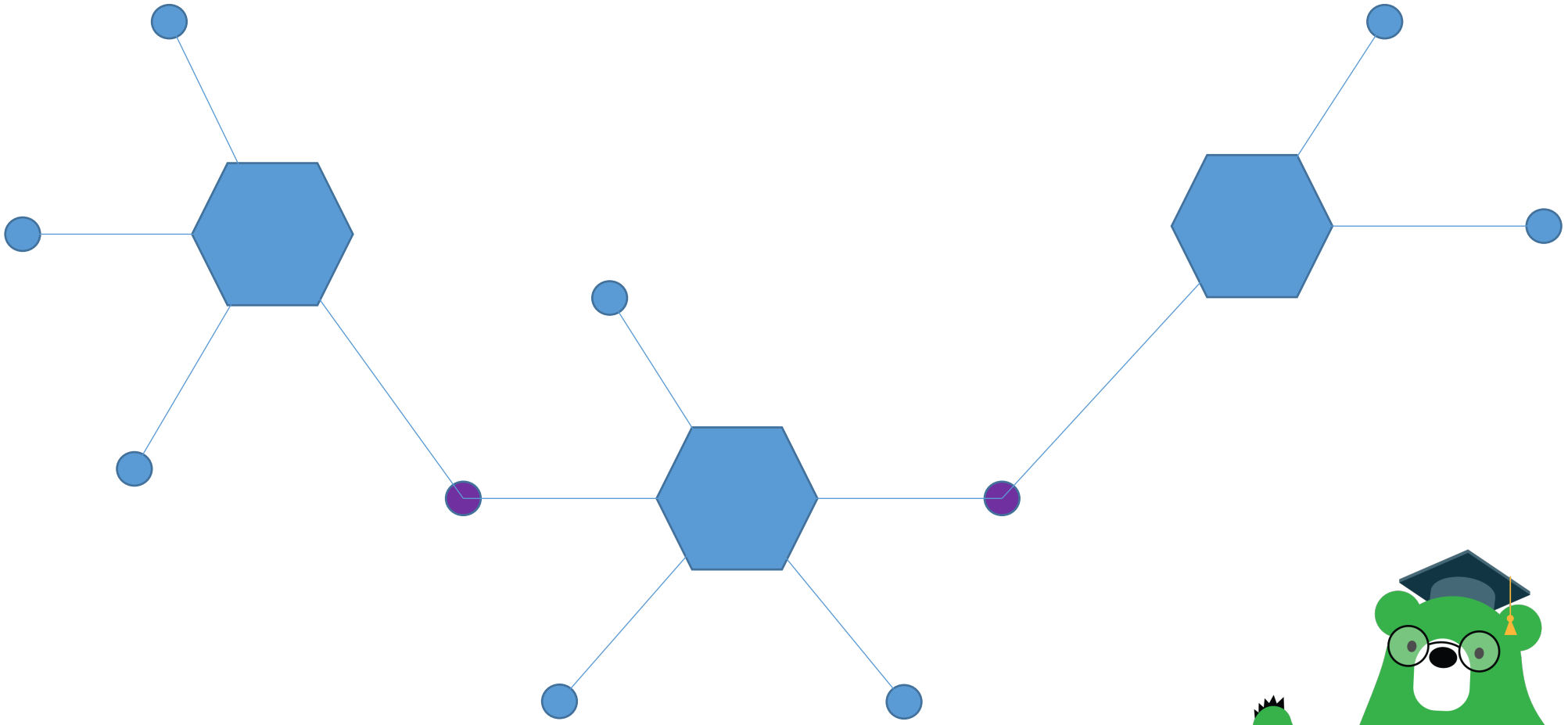


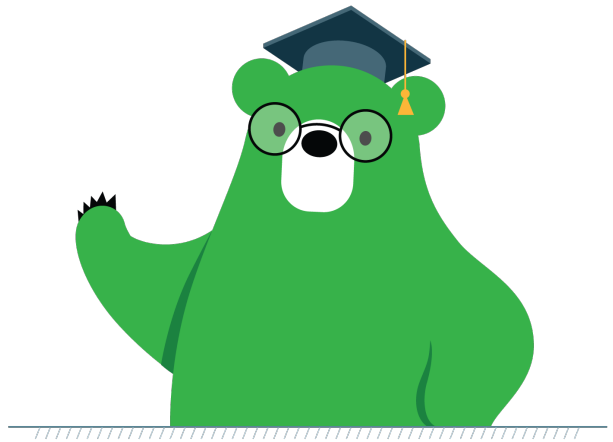
Message Broker

In General

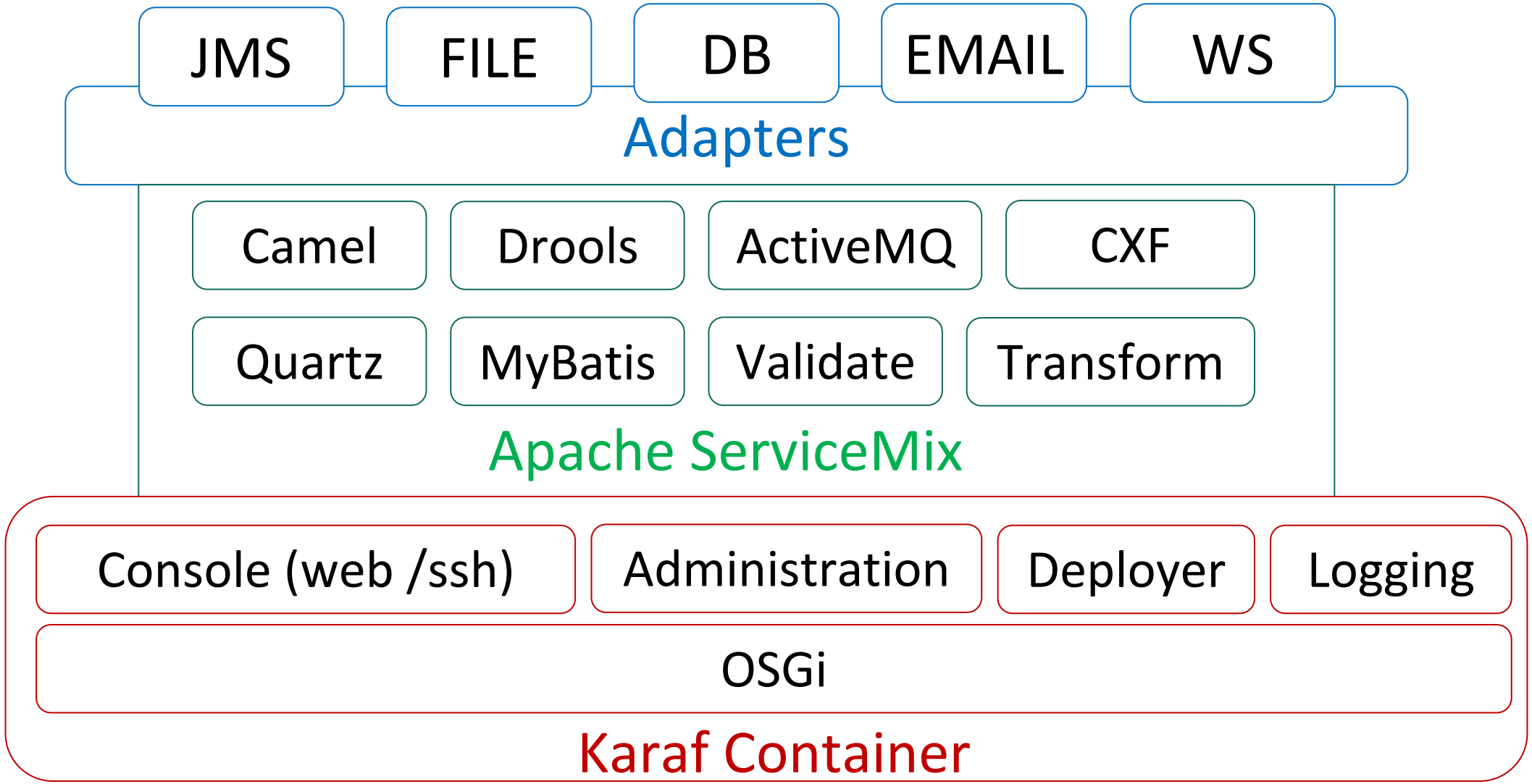


Message Broker

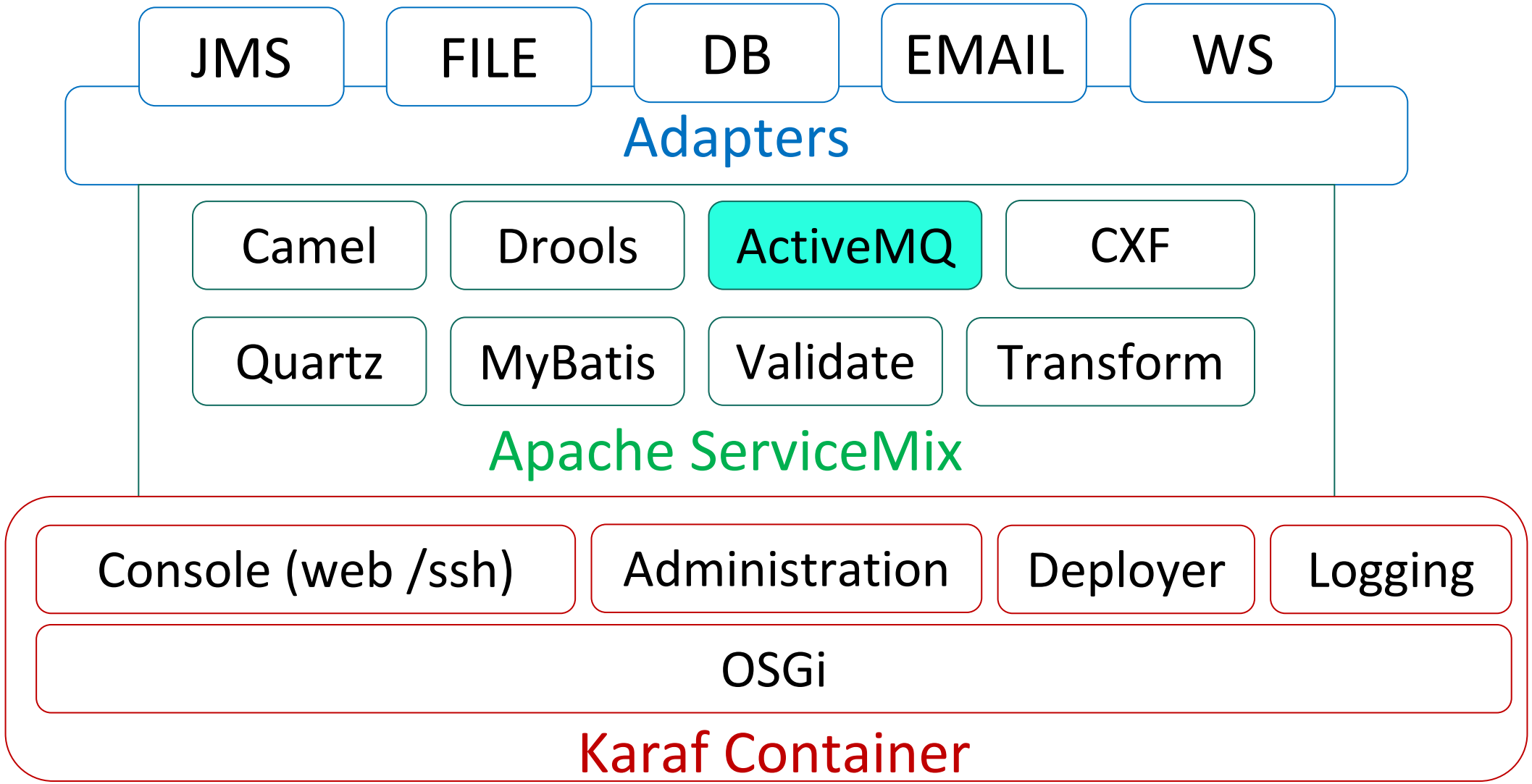




Enterprise Service Bus - ServiceMix



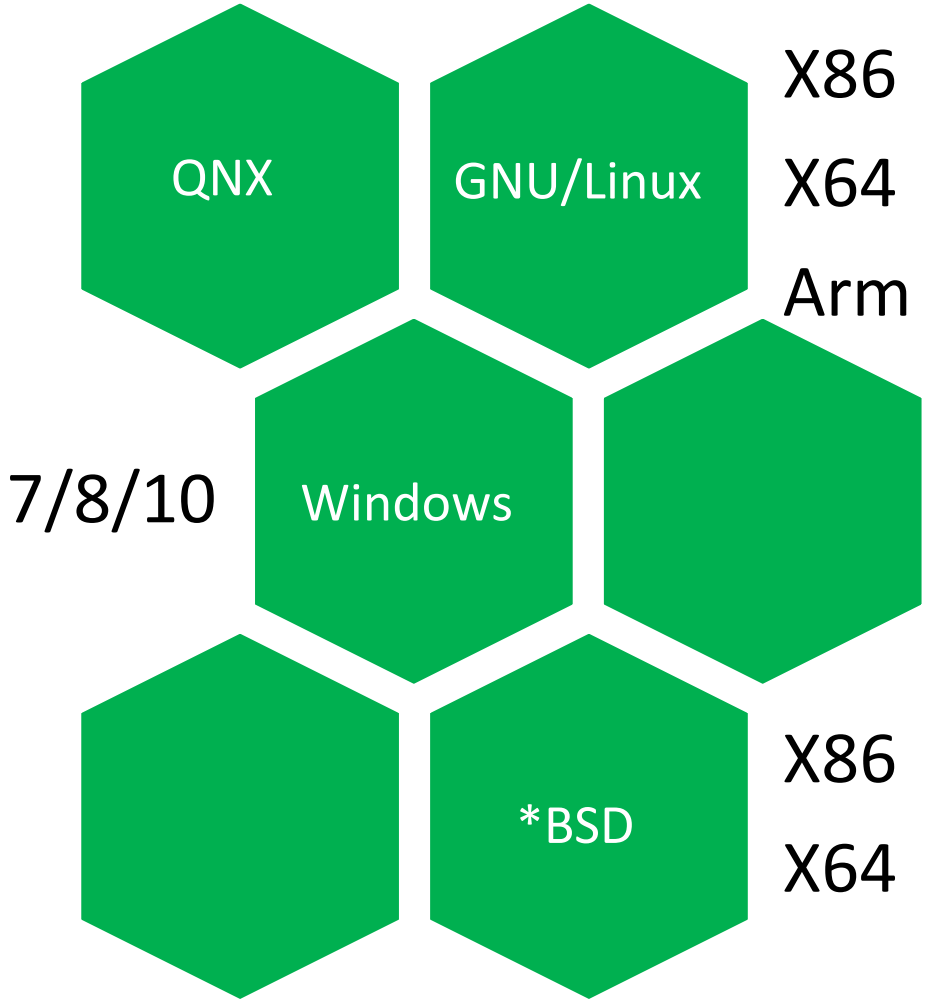
Enterprise Service Bus - ServiceMix



Message Broker

Почему пришлось писать свой брокер?

Message Broker

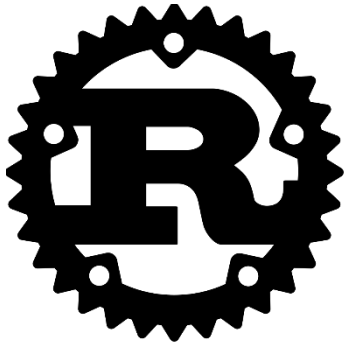


Message Broker

?

Message Broker

Message Broker



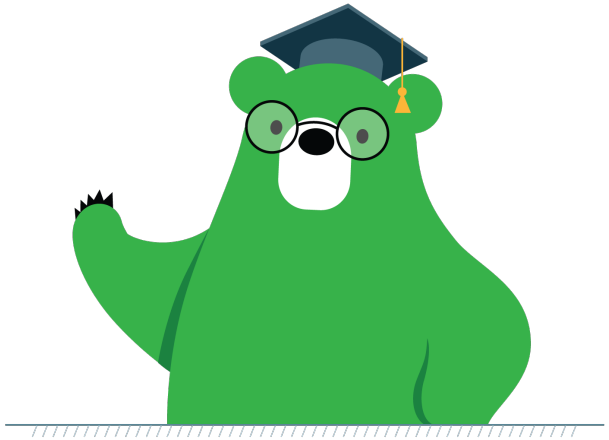
?

Импортозамещение

KASPERSKY lab

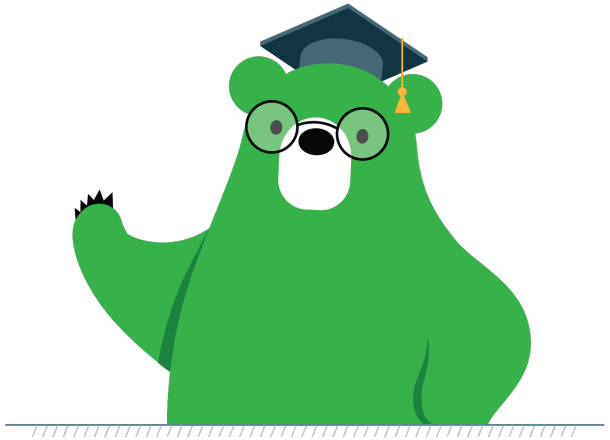
Свойства Брокера

- Управляемость
- Контролируемость
- Адаптируемость
- Интегрируемость



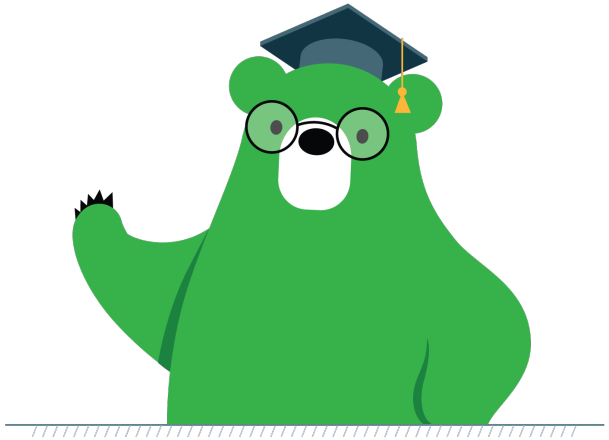
Управляемость

- Оркестровка
- Файлы конфигурации
- Web-интерфейс
- ...



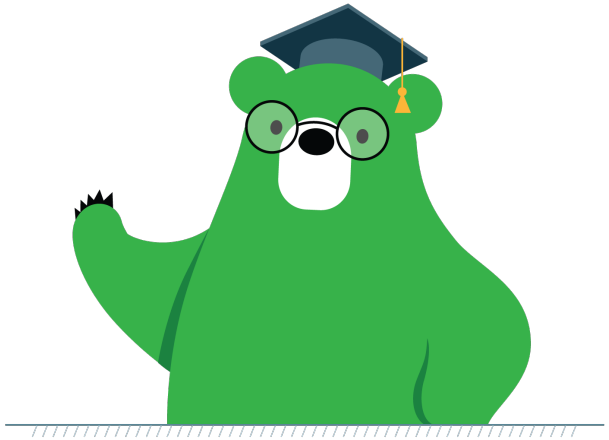
Контролируемость

- Браузер очередей
- Журналирование «всего»
- Счетчики
- …

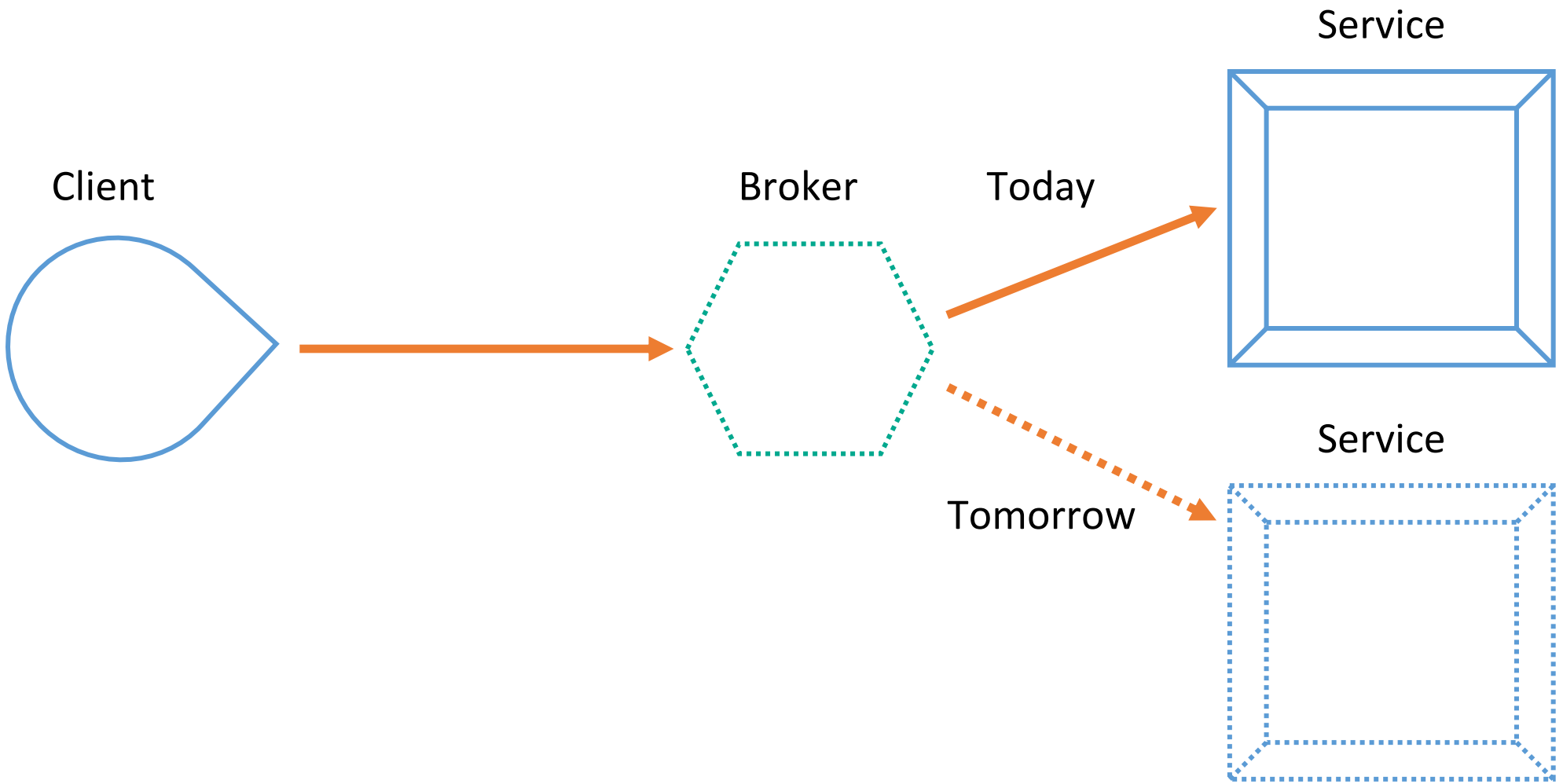


Адаптируемость

- Трансформация сообщений
- Маршрутизация сообщений
- ...



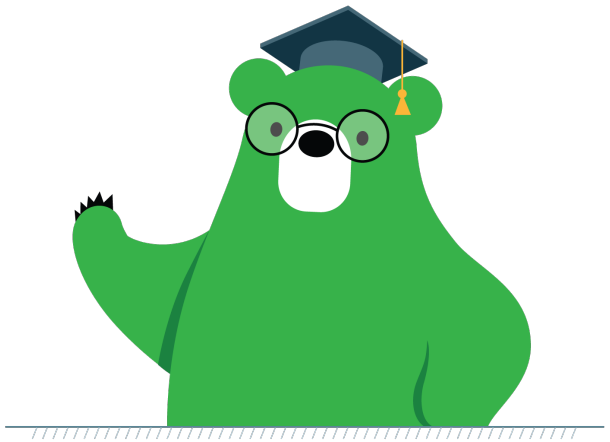
Адаптируемость [пример]



Интегрируемость

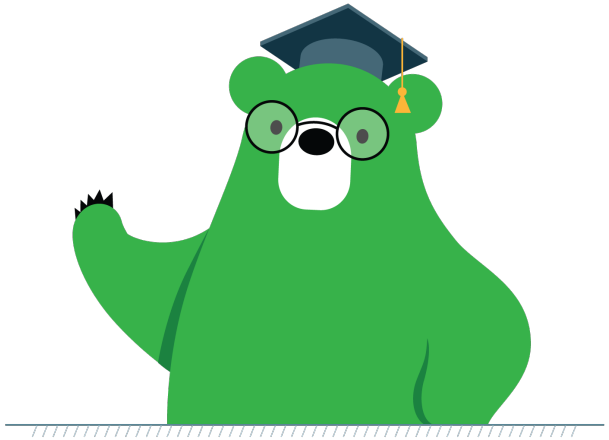
- Java Message Service (JMS 1.1 и/или 2.0. JMSCTC)
- Протоколы (AMQP, MQTT…)
- Multi Language API
- …

http://jmscts.sourceforge.net/



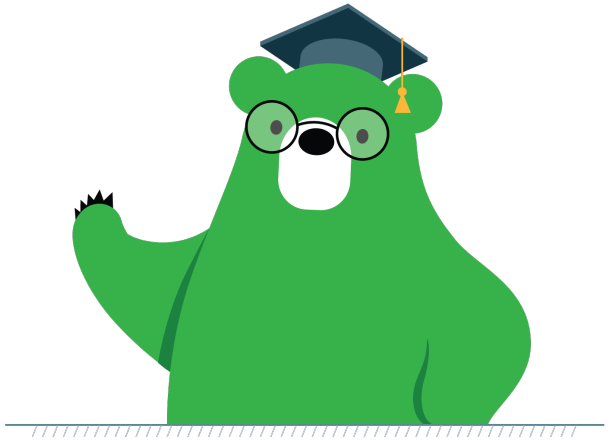
Протоколы

- AMQP
- MQTT
- ...

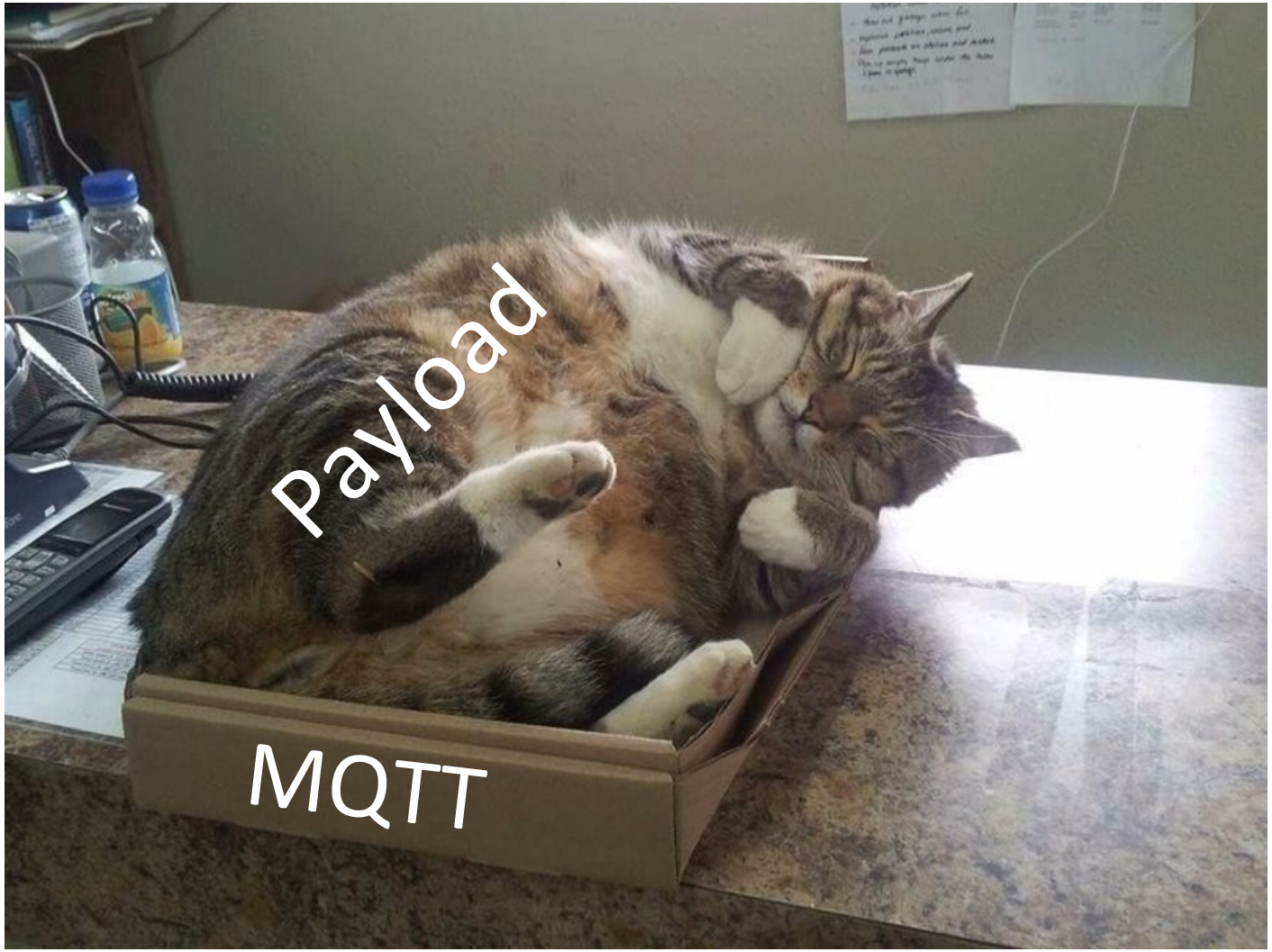


Протоколы

AMQP 0.9 != AMQP 1.0



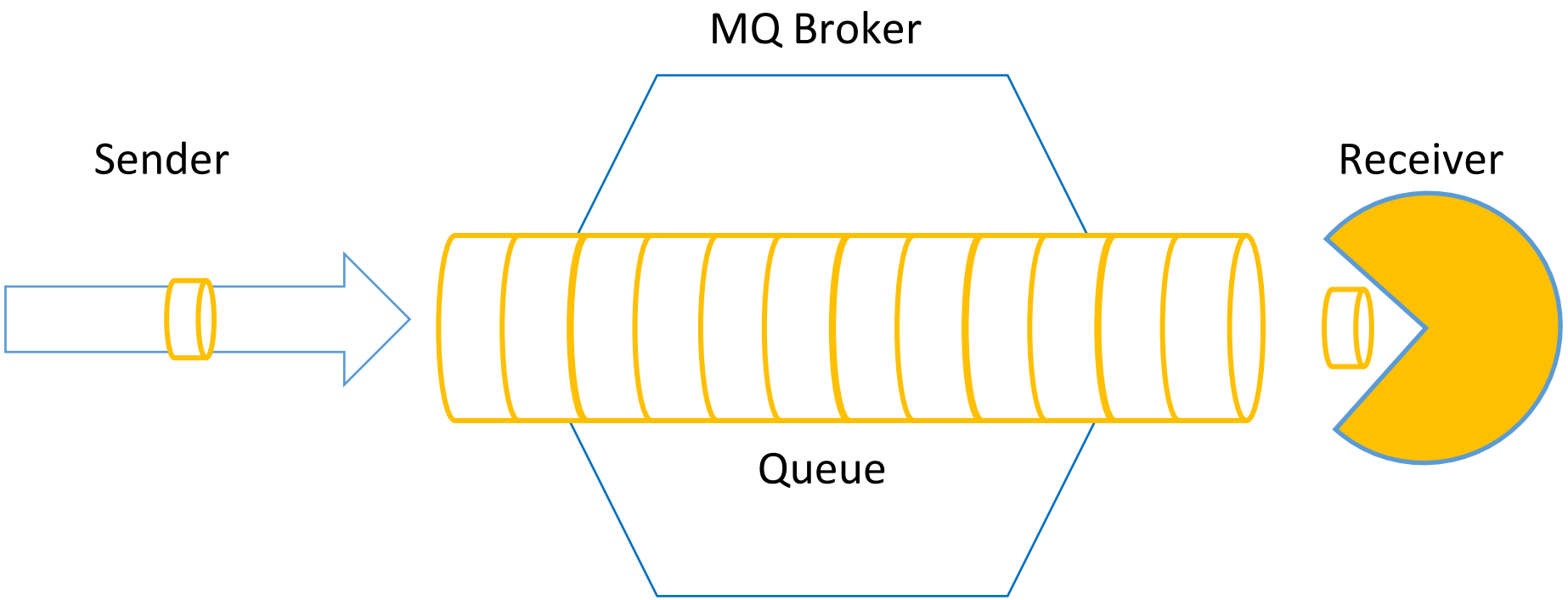
Протоколы



Основные сценарии обмена сообщениями

Message Broker

Point to Point



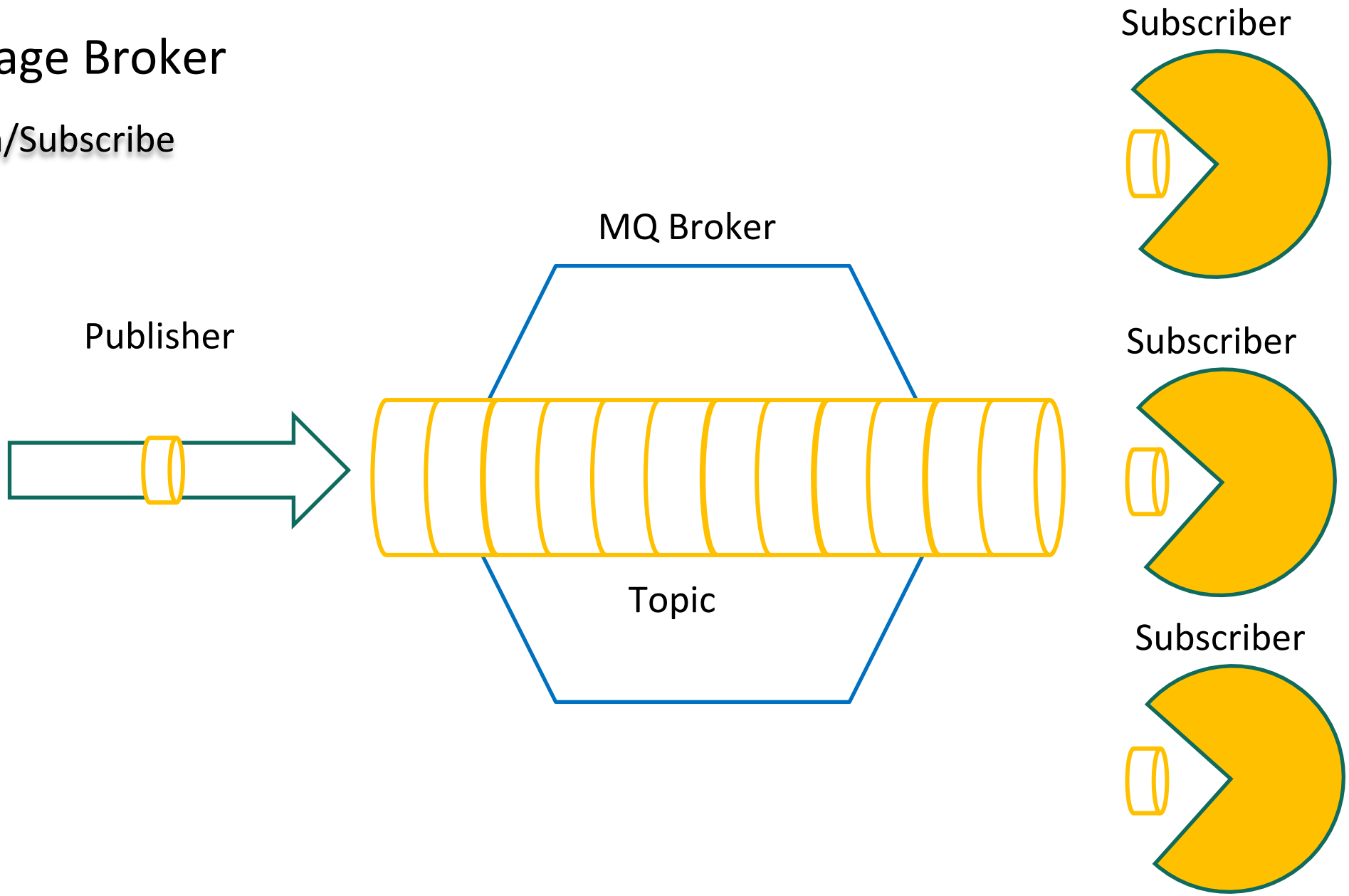
Message Broker

Point to Point

```
// ...
producer->send(message);
// ...
cms::Message *message = consumer->receive();
// ...
```

Message Broker

Publish/Subscribe



Message Broker

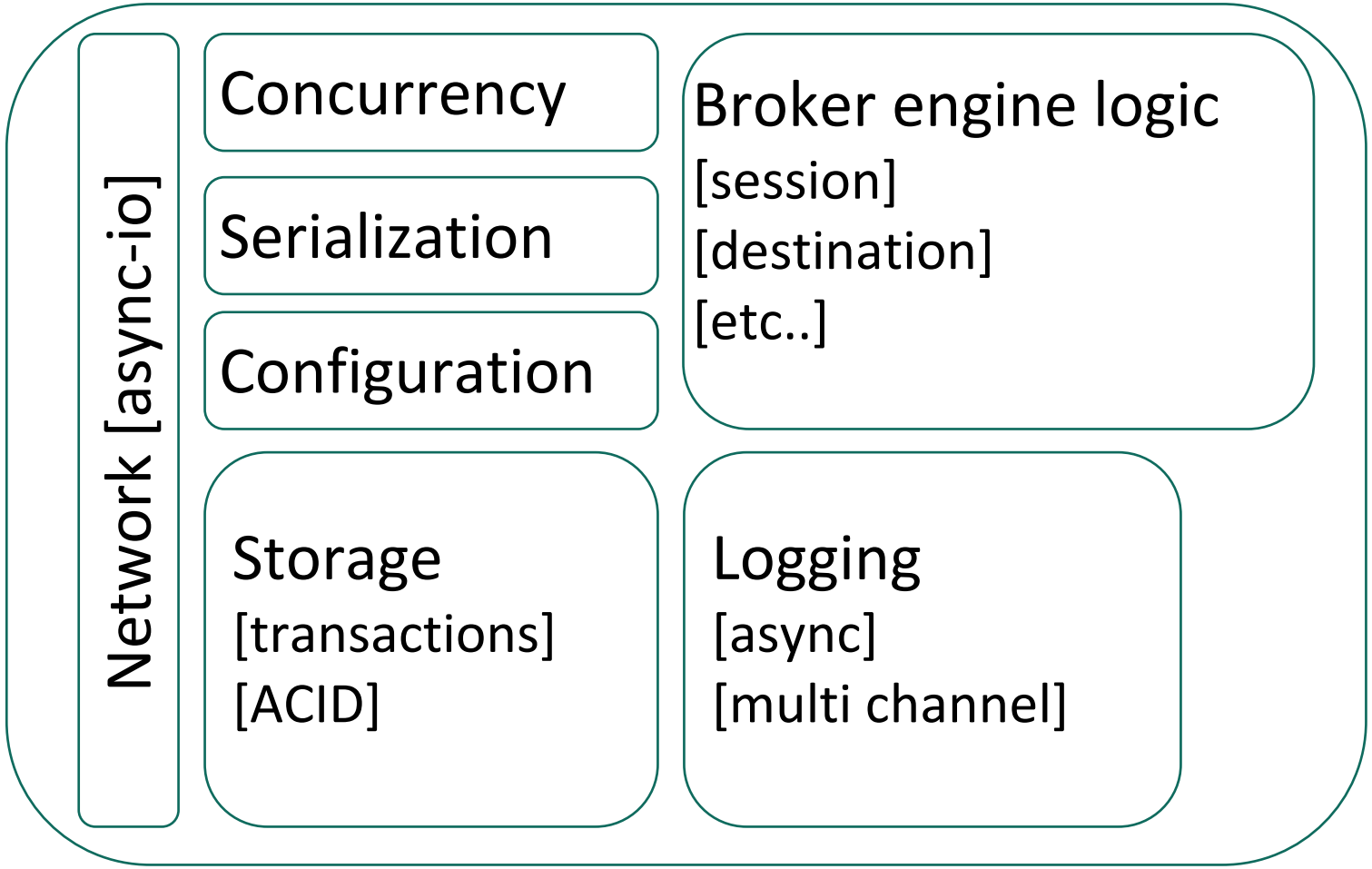
Publish/Subscribe

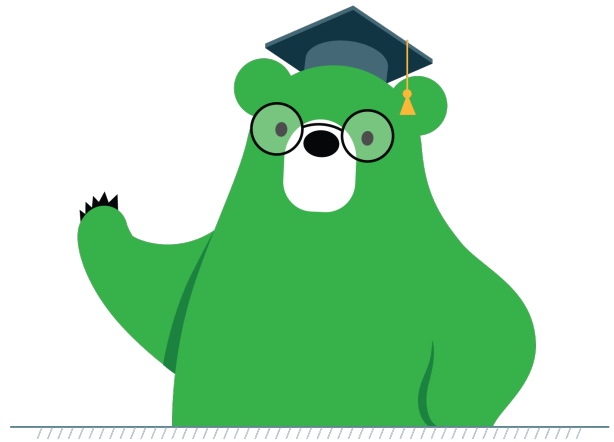
```
// …
publisher->send(message);
// …
cms::Message *message = subscriber1->receive();
cms::Message *message = subscriber2->receive();
cms::Message *message = subscriber3->receive();
```

Долго еще?

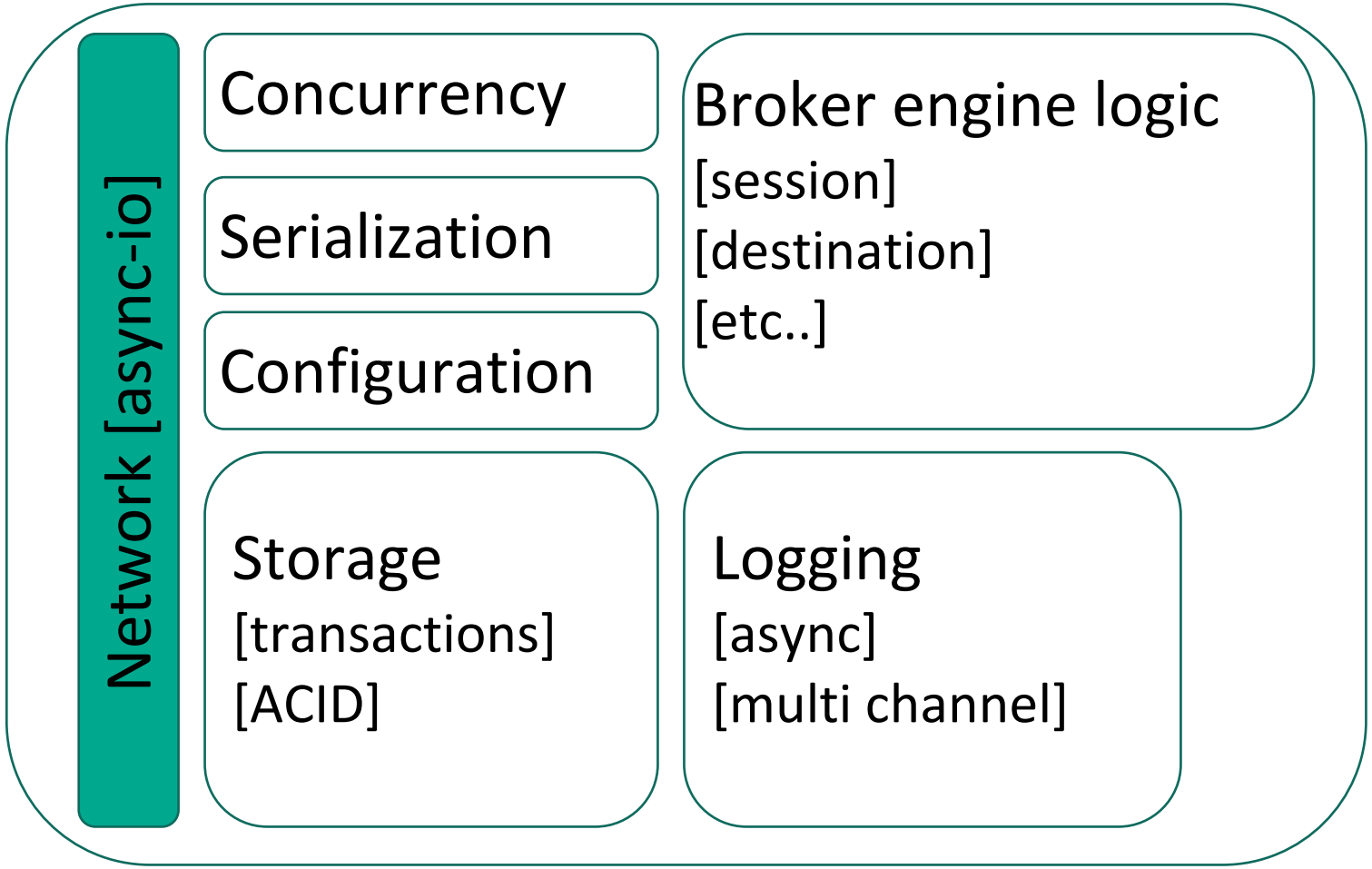
Остальное ищите в JMS спецификации...

Message Broker - Архитектура



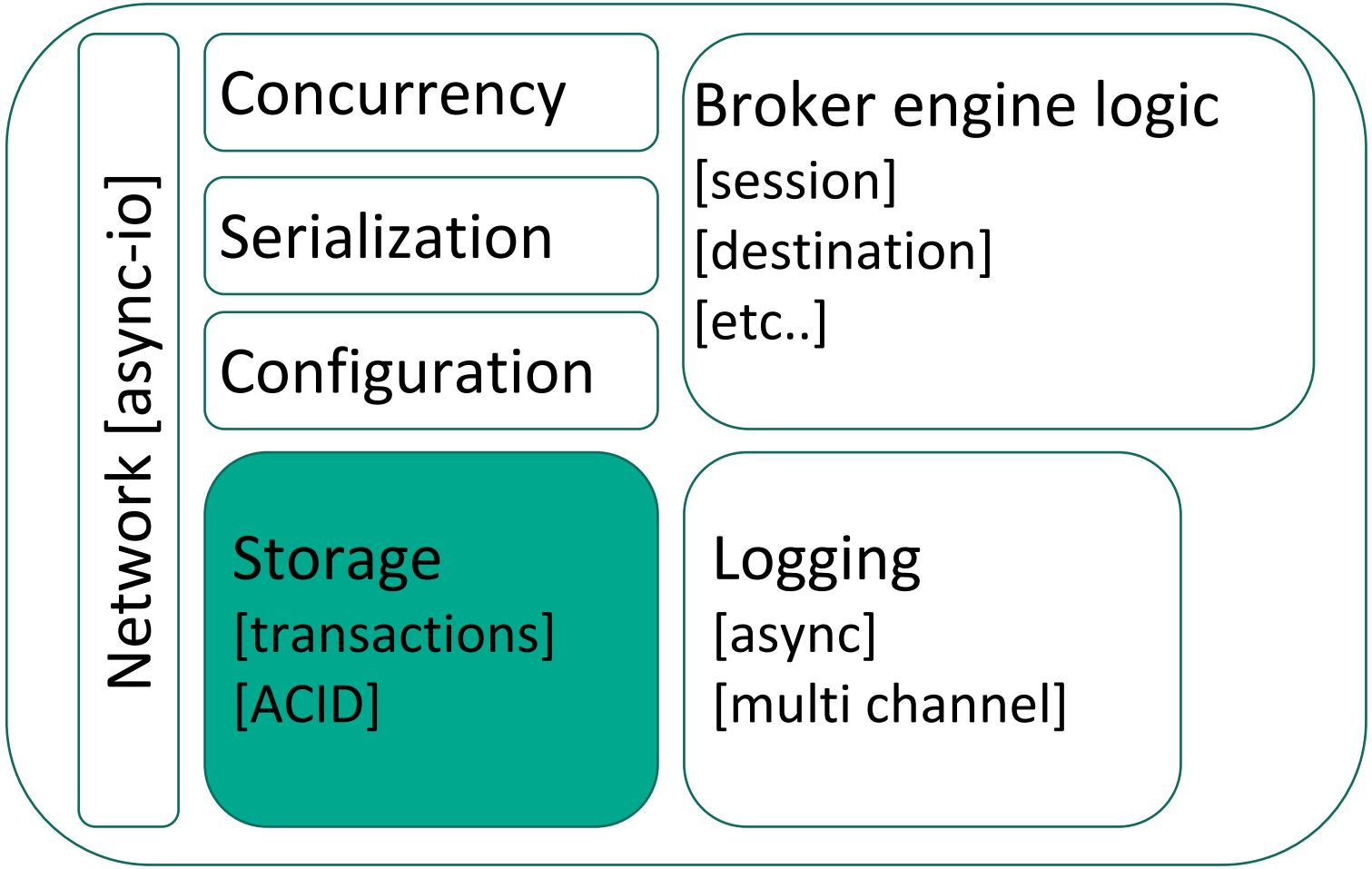


Message Broker - Архитектура



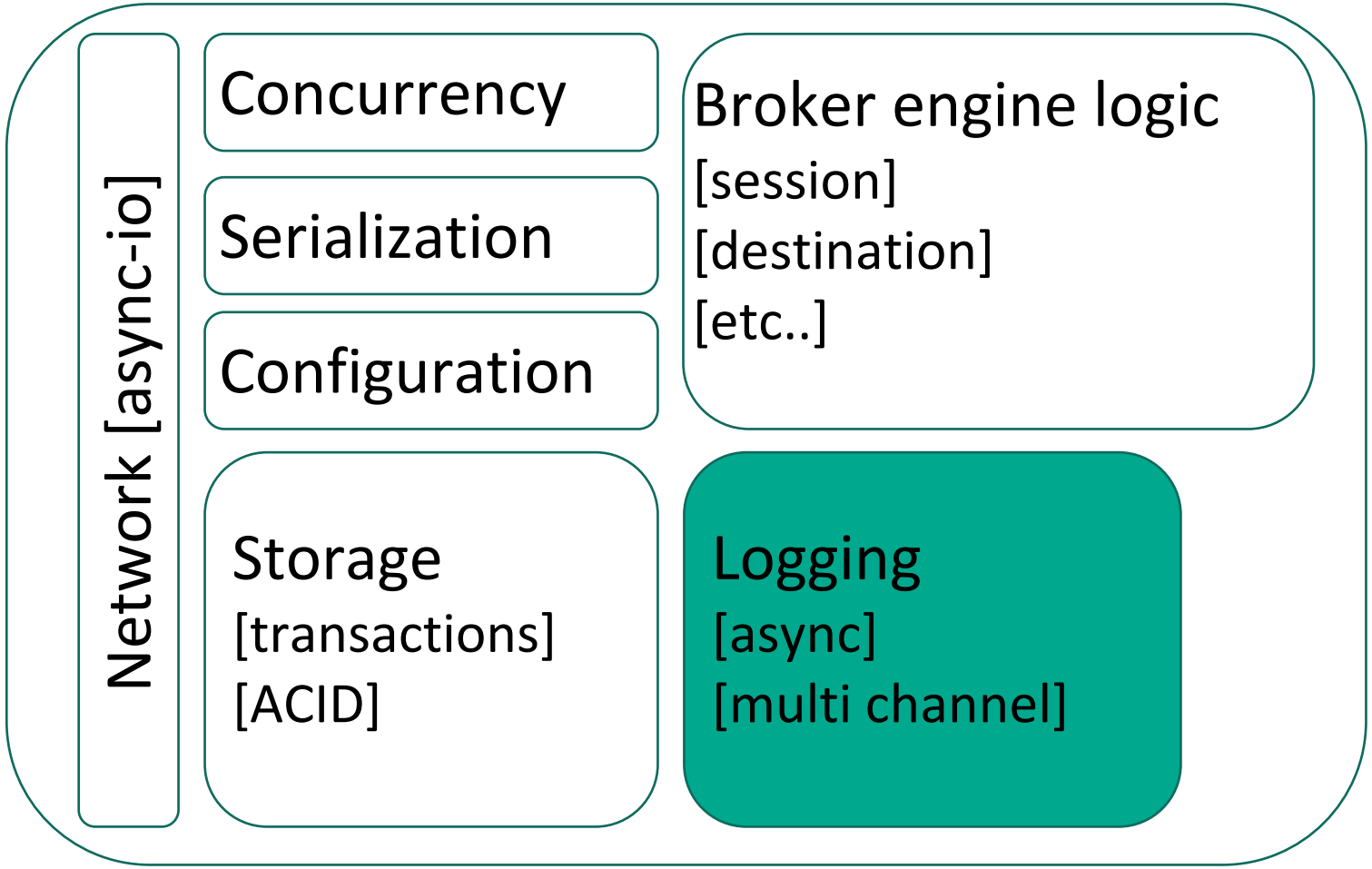
Reactor

Message Broker - Архитектура



SQL ?

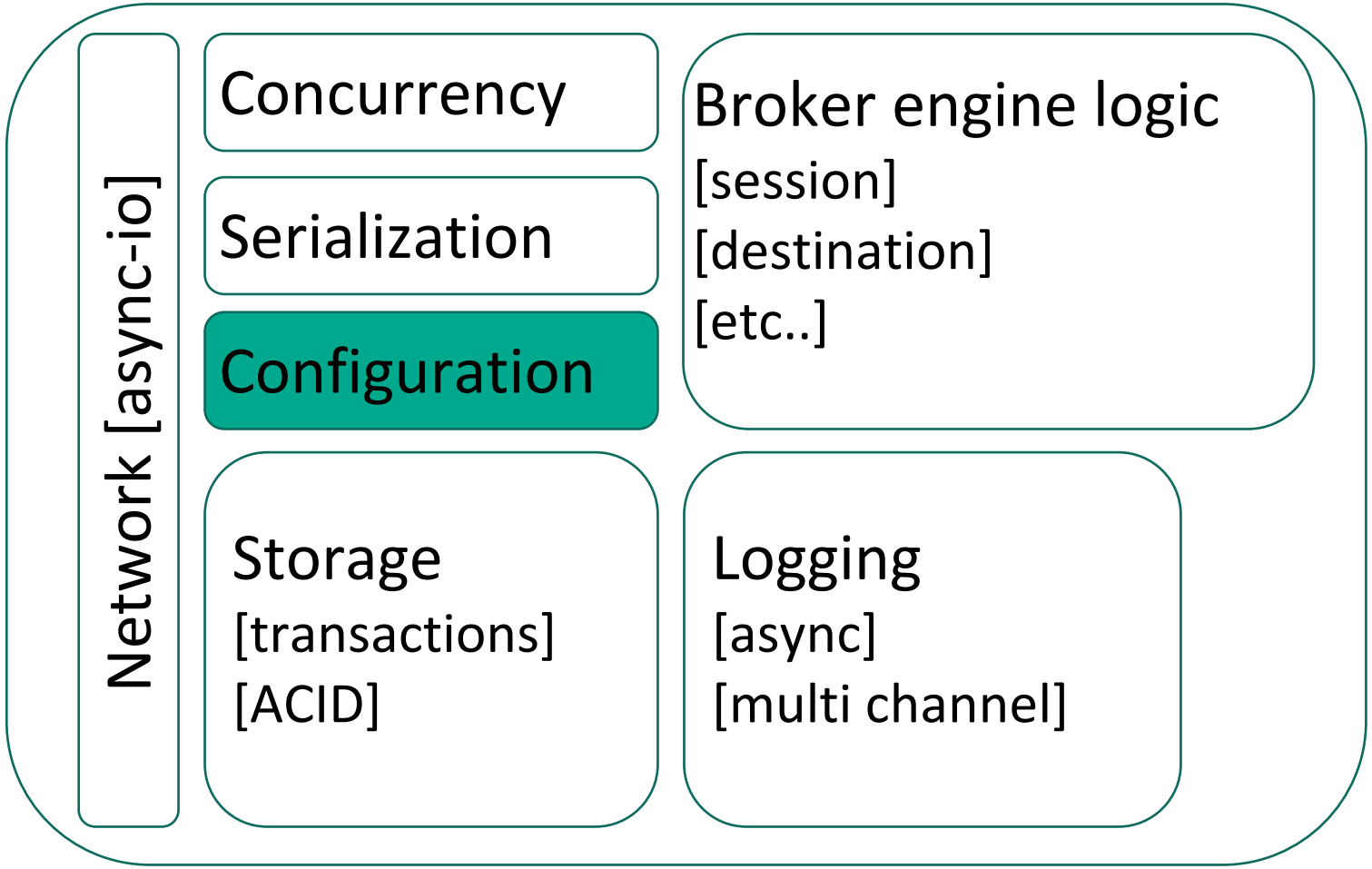
Message Broker - Архитектура



More information

Less waiting

Message Broker - Архитектура

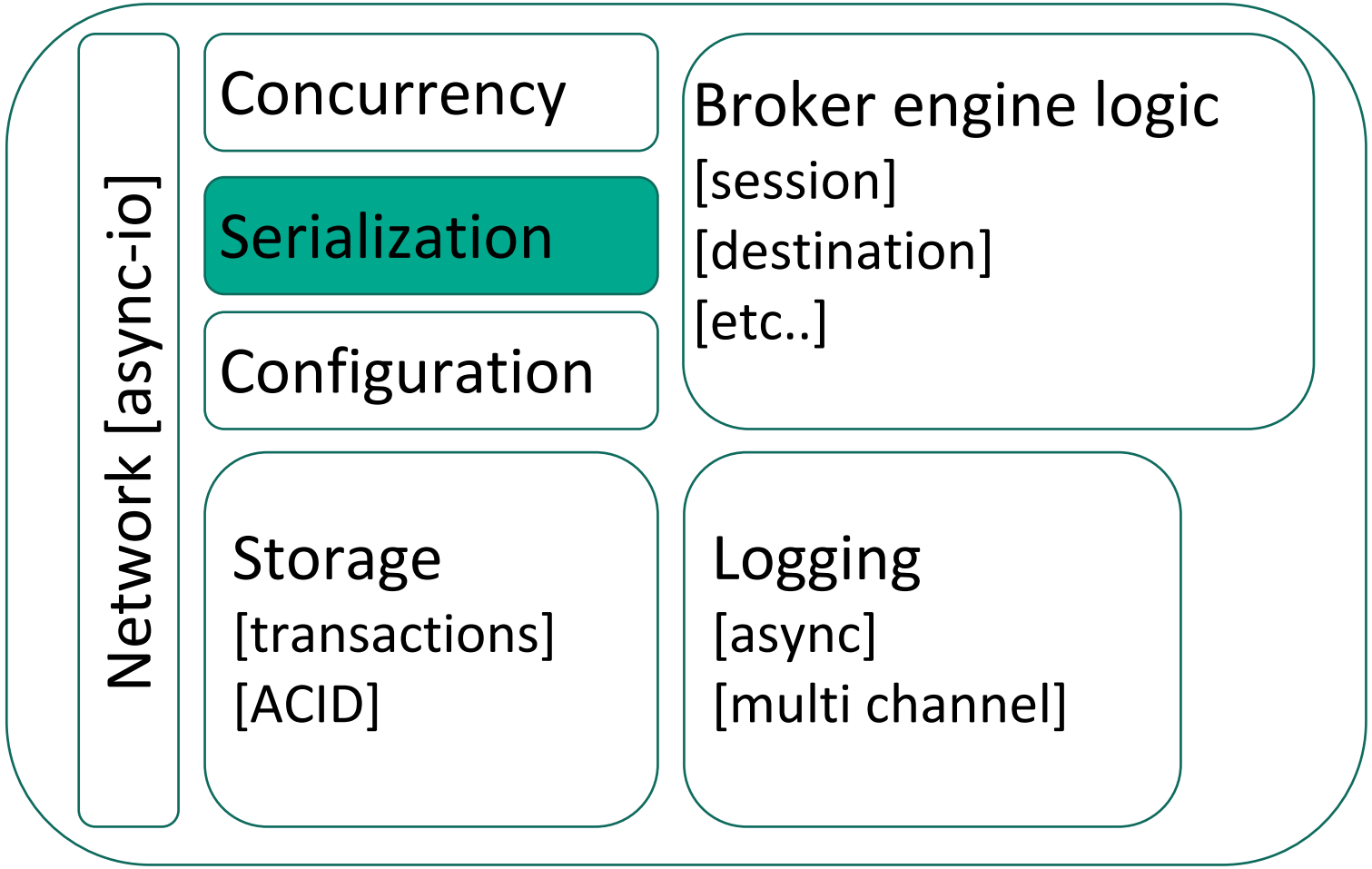


XML

JSON

...

Message Broker - Архитектура



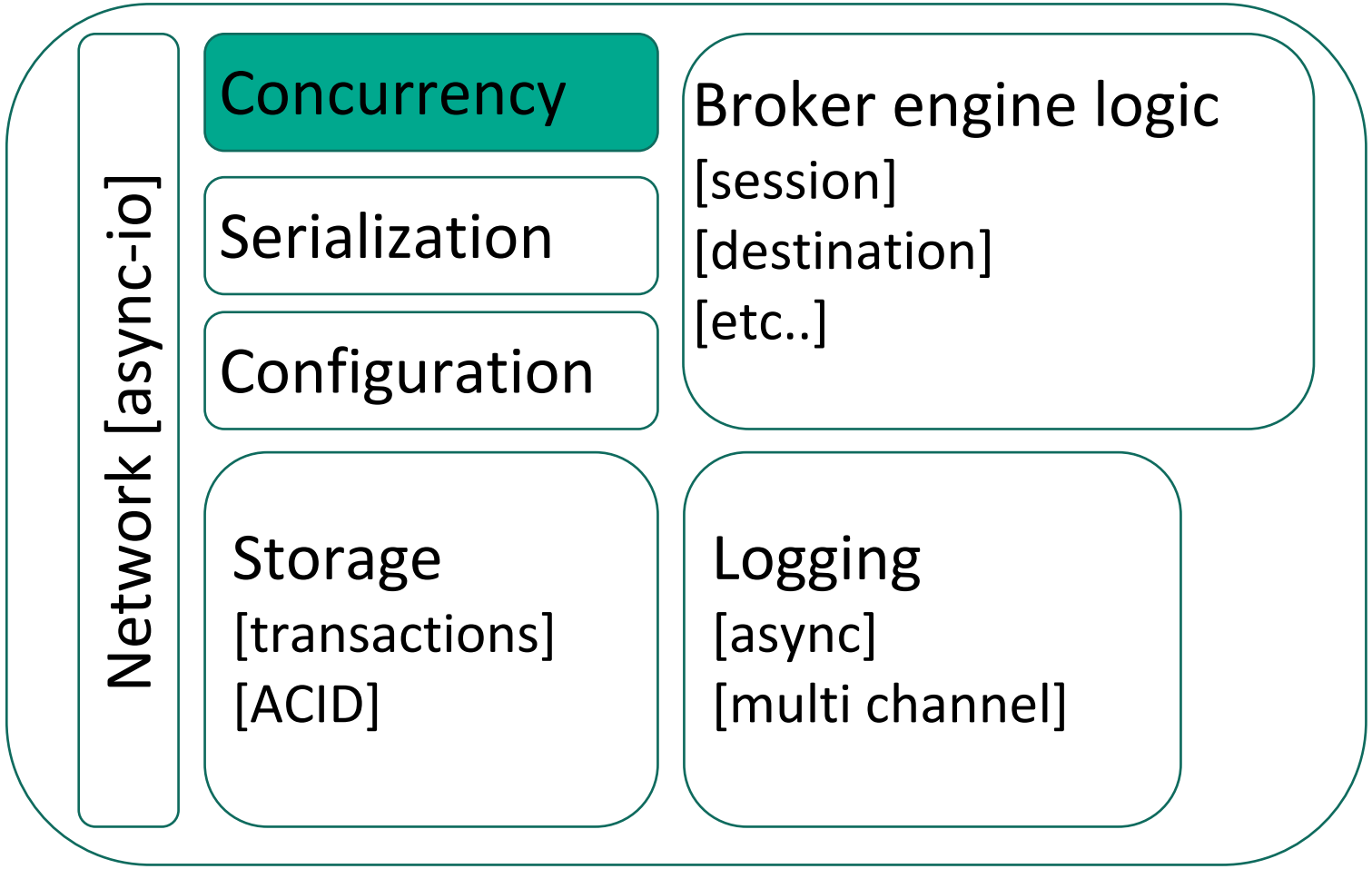
Protobuf

Message Broker - Архитектура

Protobuf

```
message ProtoMessage {
   oneof ProtoMessageType {
      Connect connect = 1;
      Connected connected = 2;
      Disconnect disconnect = 3;
      ClientInfo client_info = 4;
      Session session = 5;
      // ...
      Undestination undestination = 25;
   }
   required string object_id = 100;
   required int32 request_reply_id = 101;
}
```

Message Broker - Архитектура



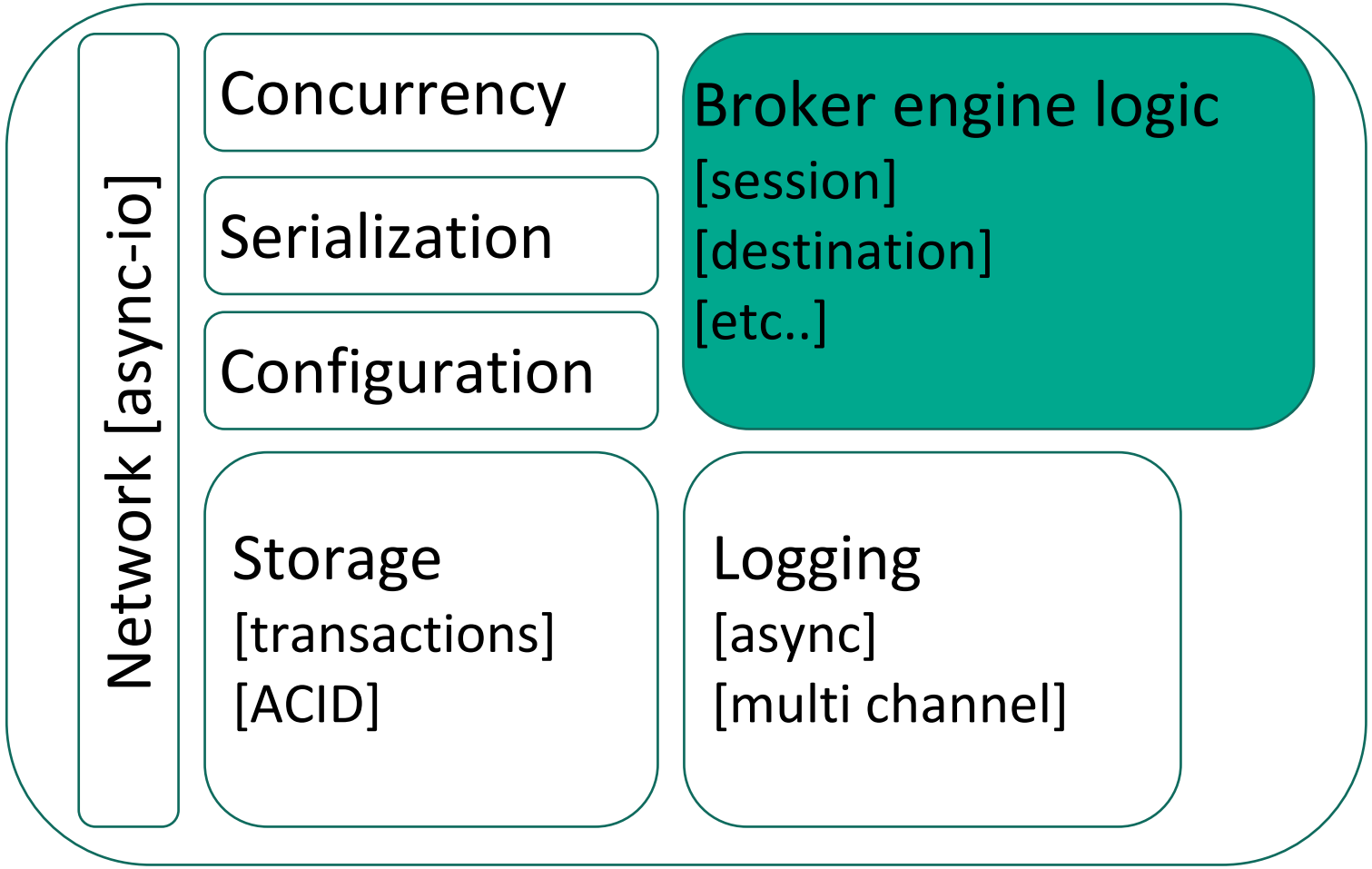
Wait-Free

Lock-Free

Map

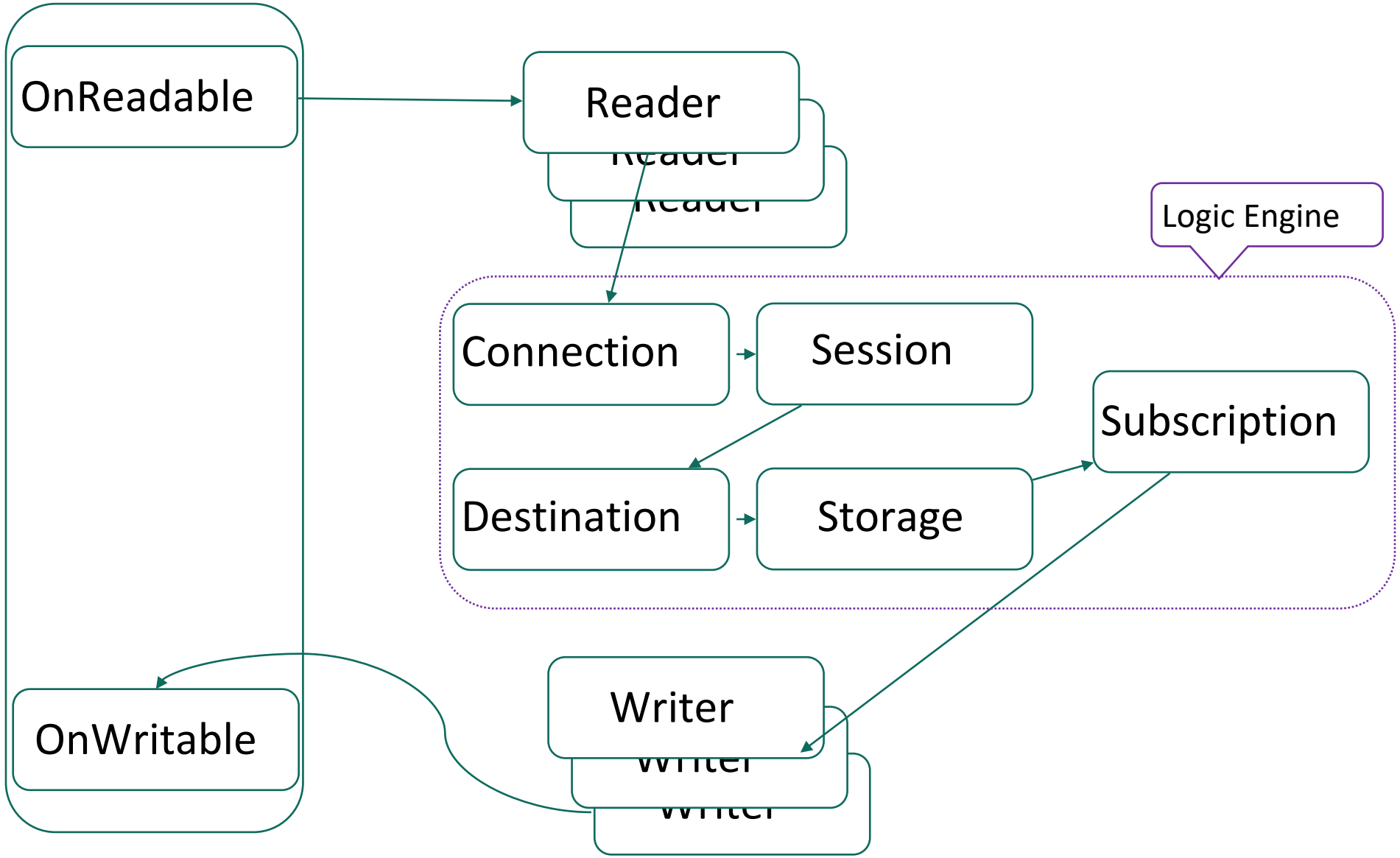
Queue

Message Broker - Архитектура

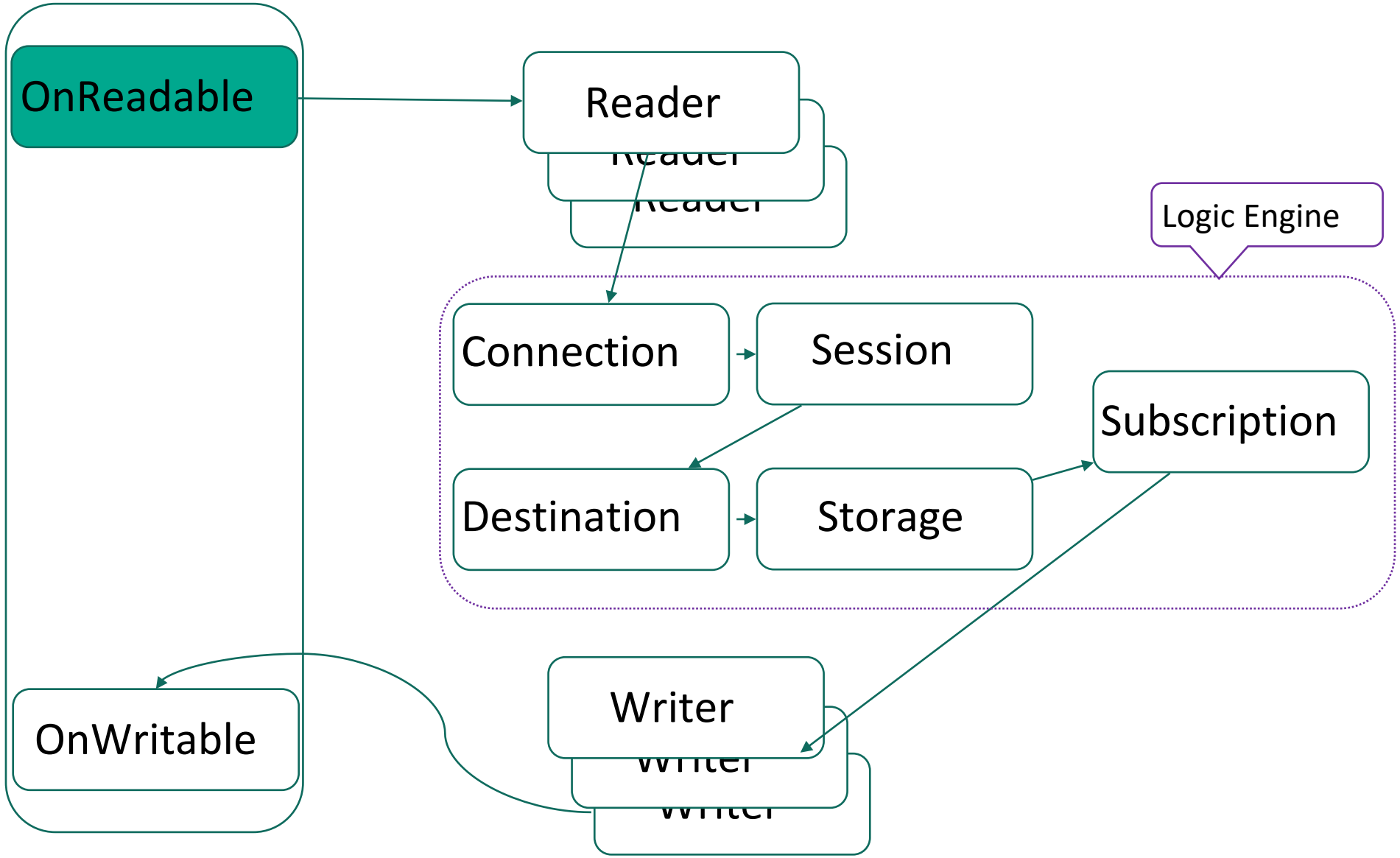


JMS

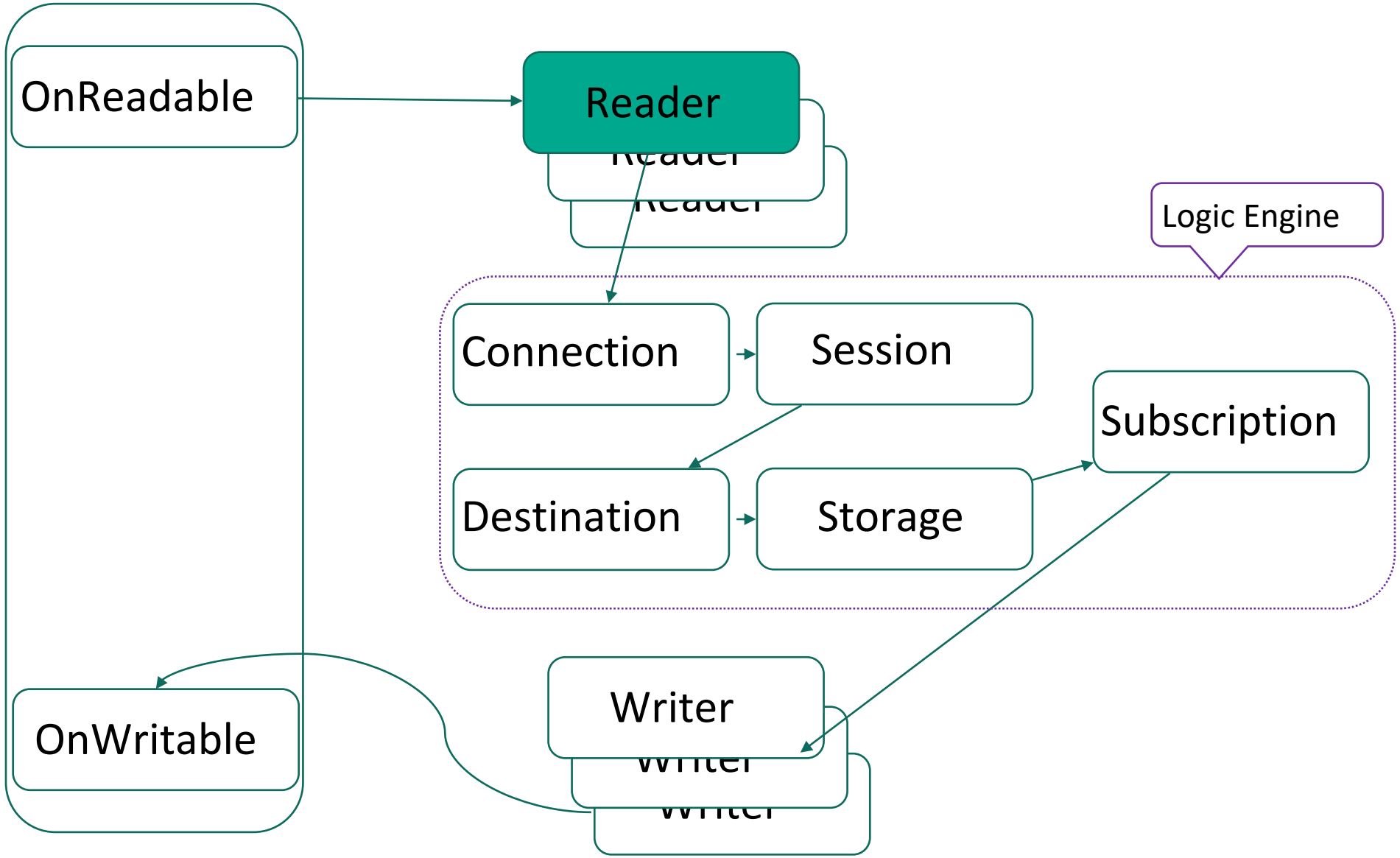
Message Broker - Архитектура



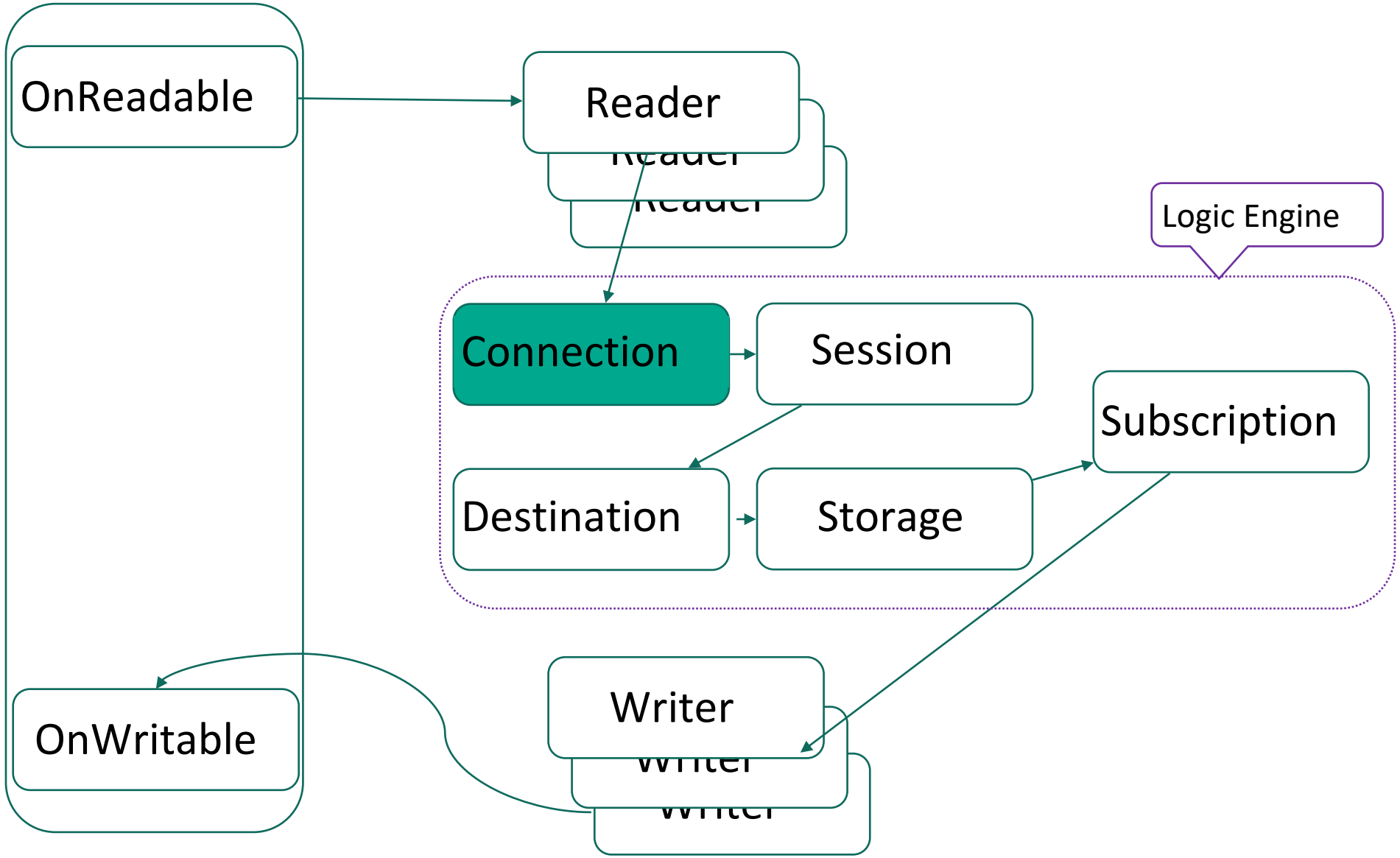
Message Broker - Архитектура



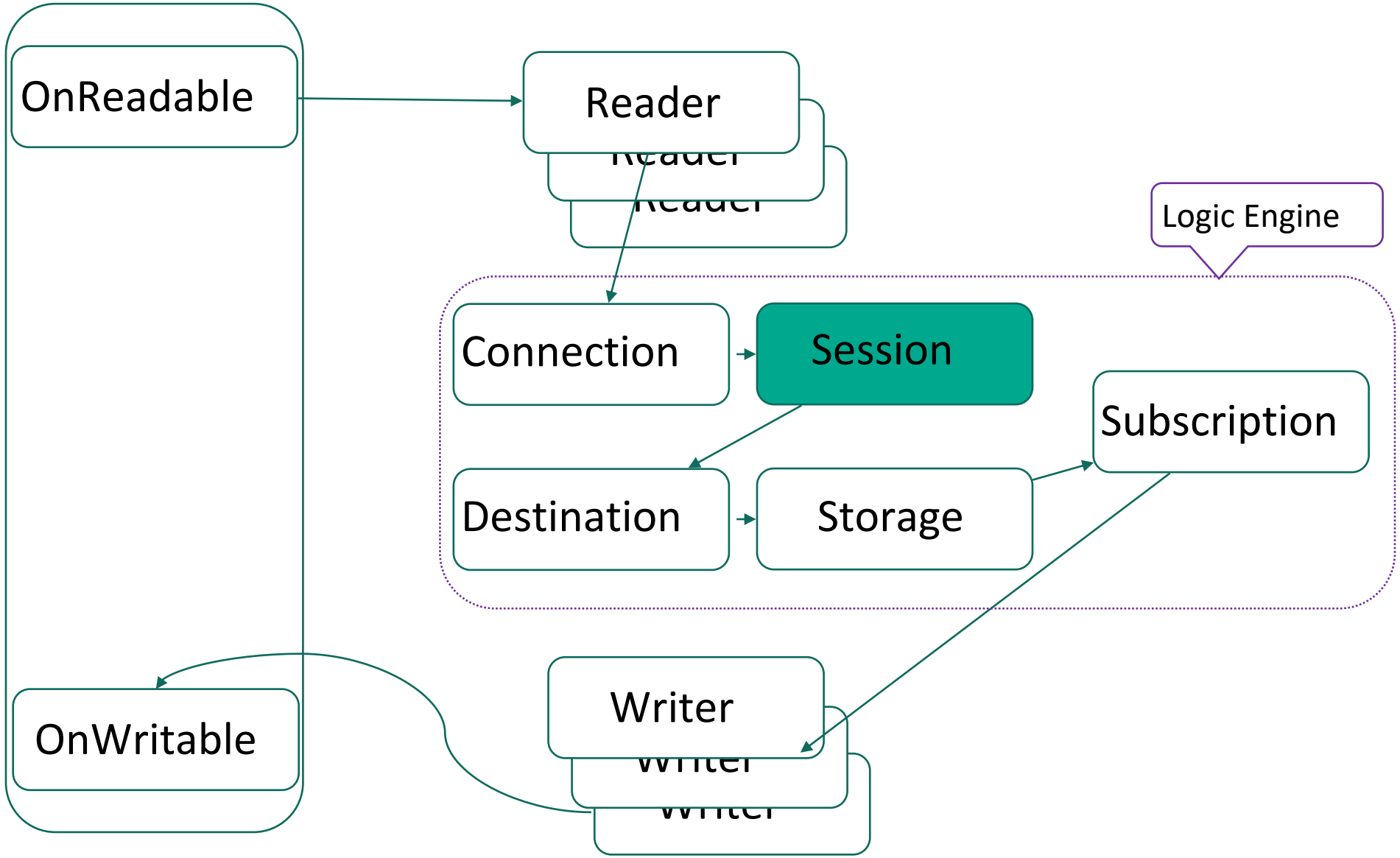
Message Broker - Архитектура



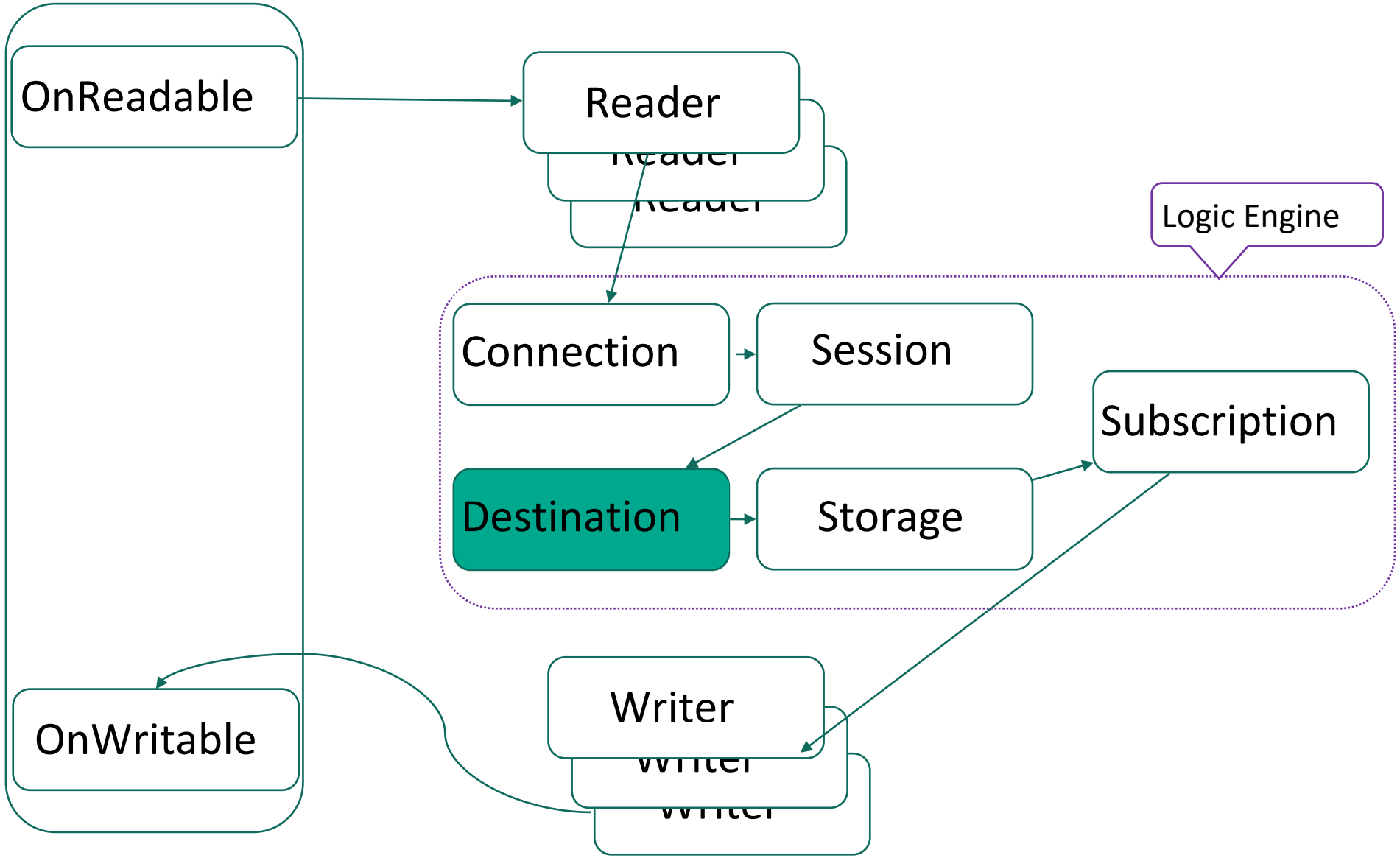
Message Broker - Архитектура



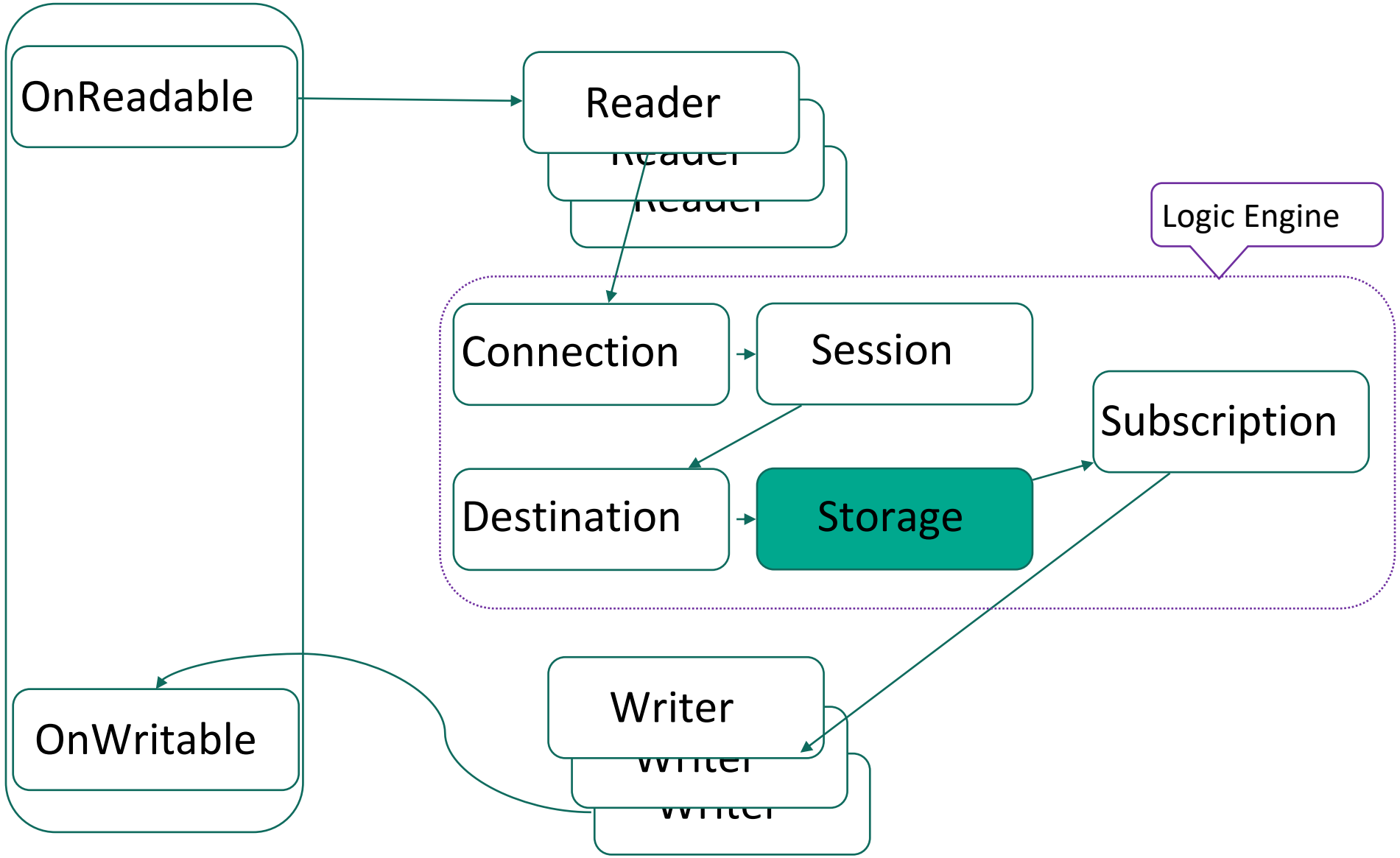
Message Broker - Архитектура



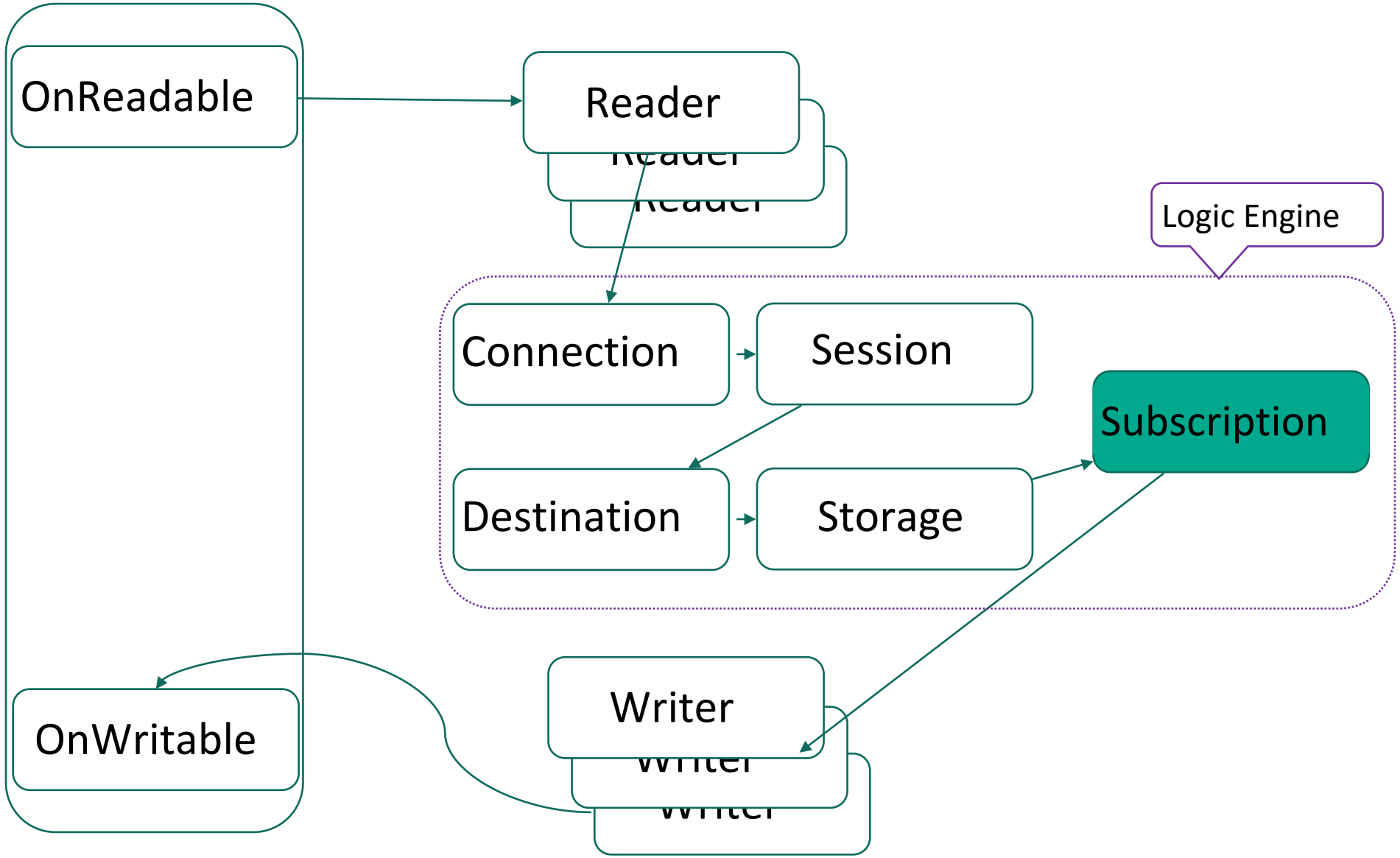
Message Broker - Архитектура



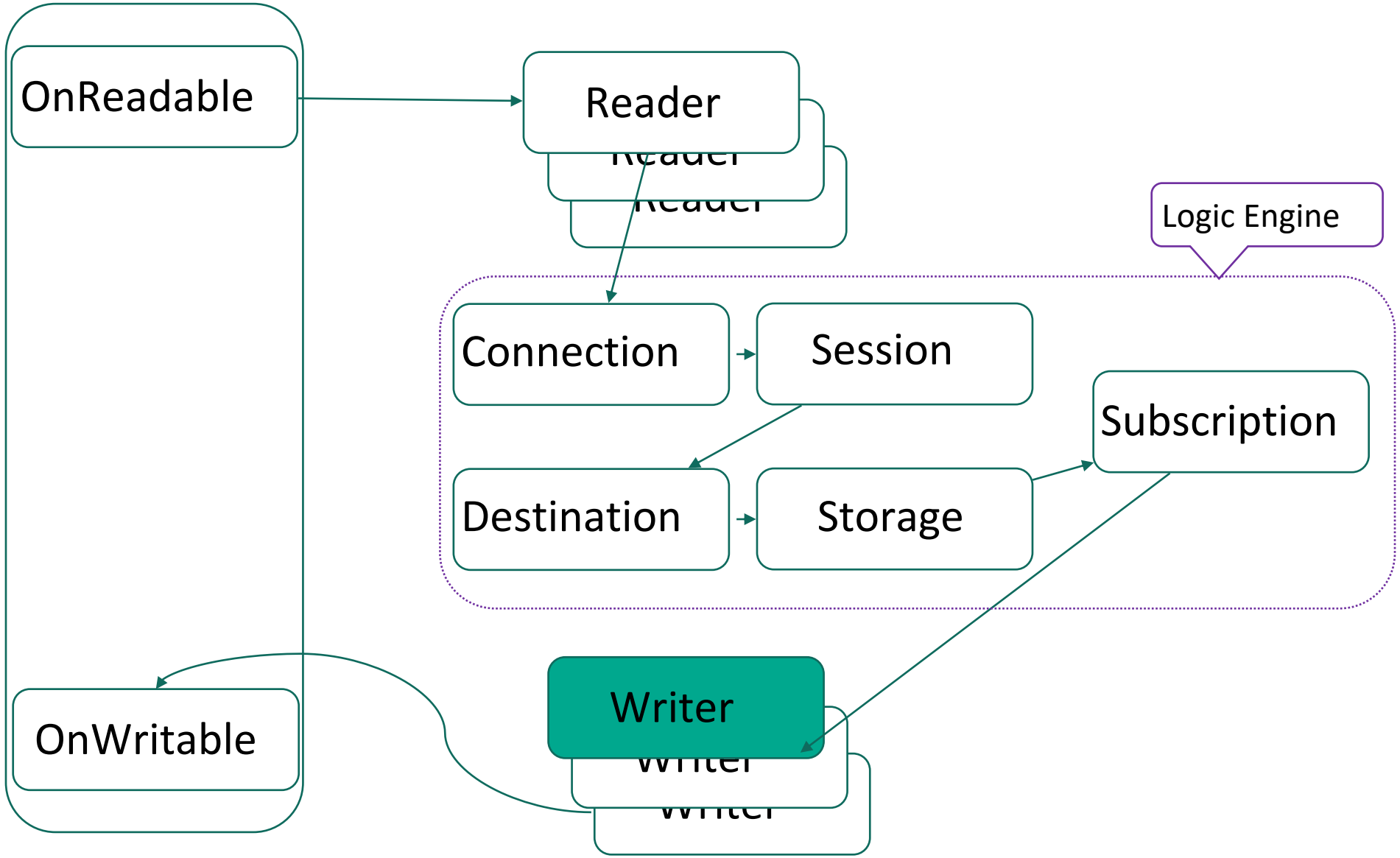
Message Broker - Архитектура



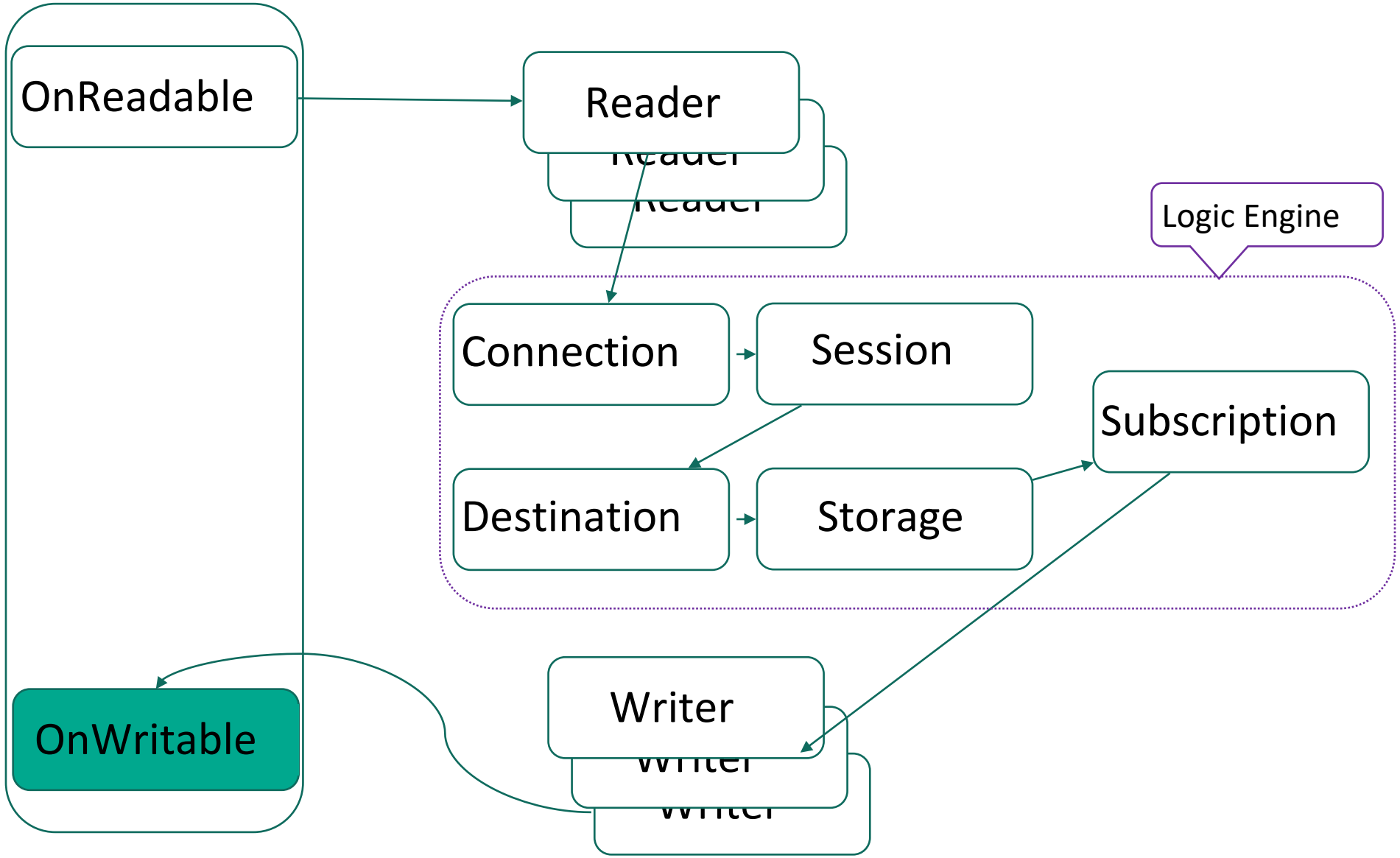
Message Broker - Архитектура



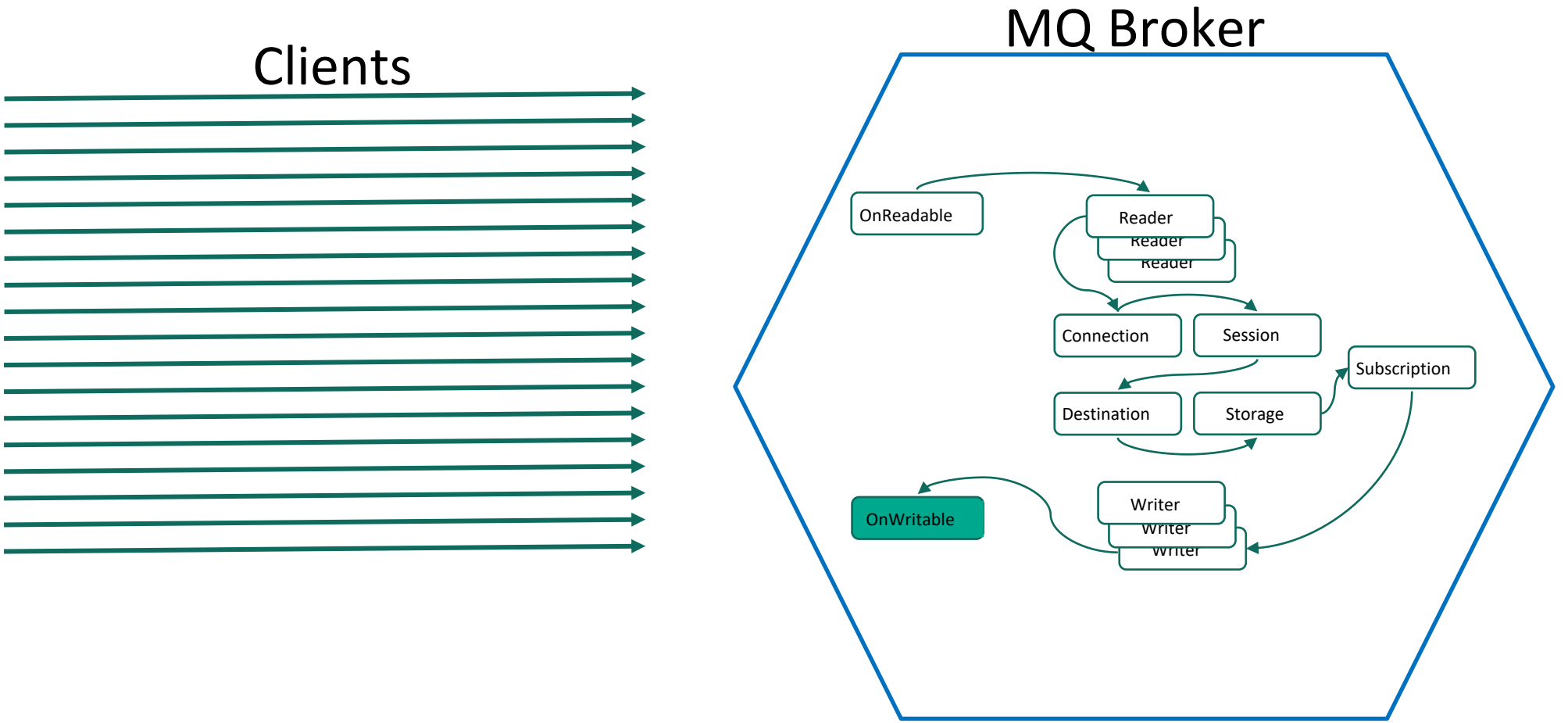
Message Broker - Архитектура



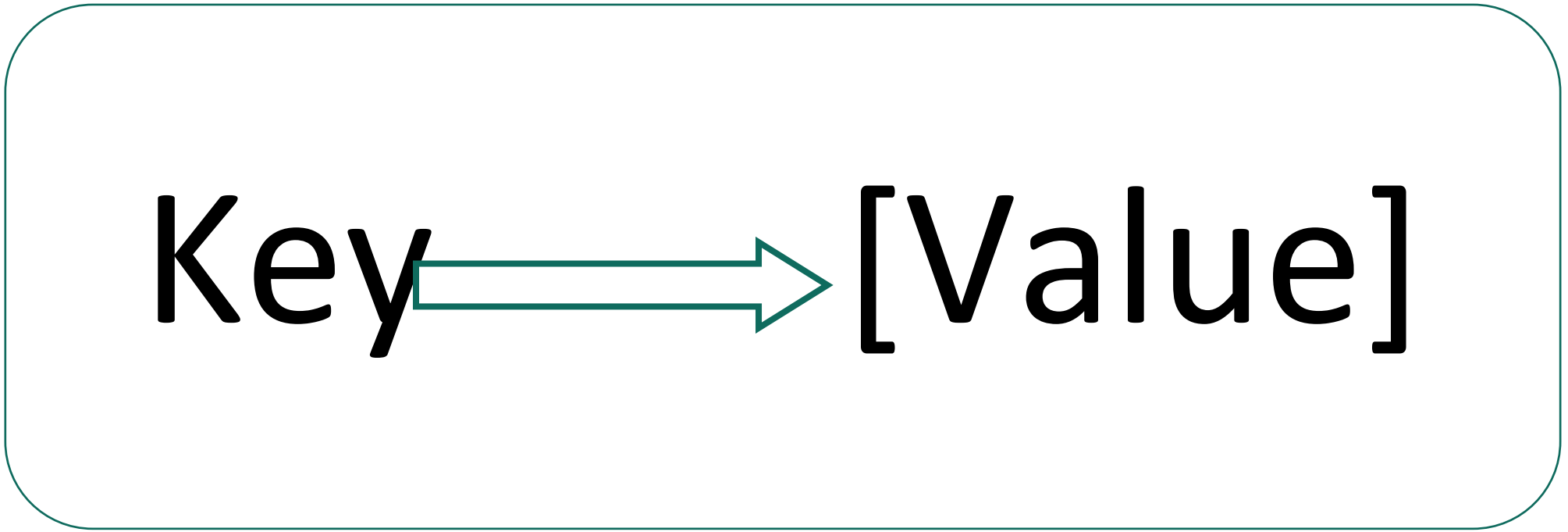
Message Broker - Архитектура



Message Broker - Архитектура

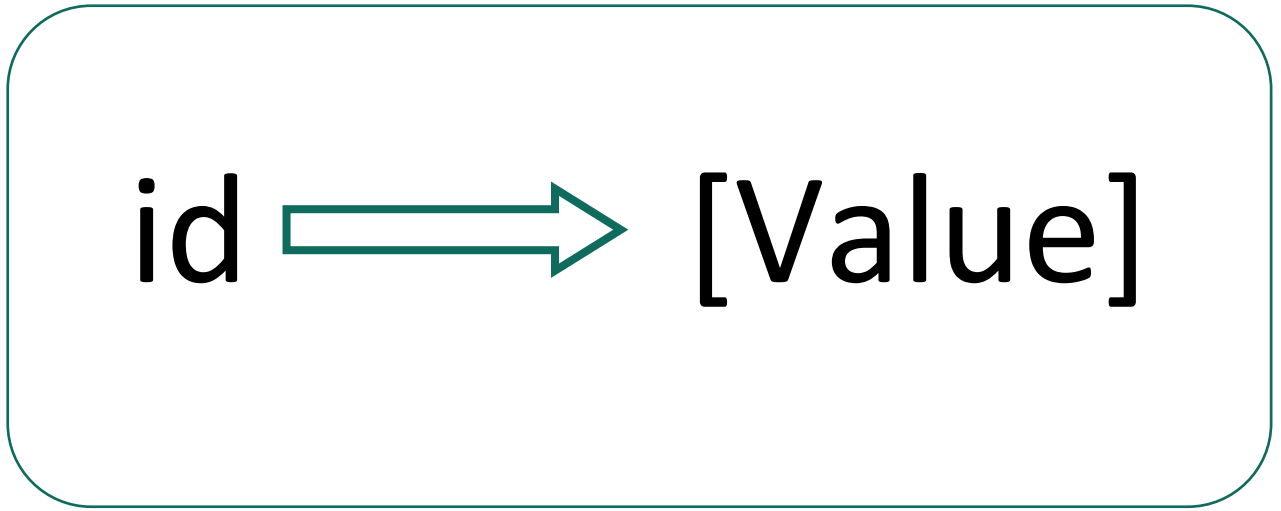


Трудности



Трудности [Concurrency]

- Connections
- Sessions
- Destinations
- Subscriptions



Трудности [Concurrency]

```
void put(std::string id, Message message) {
    ScopedWriteLock writeLock(rwLock);
    destination.emplace(std::move(id), std::move(message));
}
// ...
std::optional<Message> get(const std::string &id) {
    ScopedReadLock readLock(rwLock);
    auto it = destination.find(id);
    if (it == destination.end()) {
        return {};
    }
    return std::optional<Message>(it->second);
}
```

Трудности [Concurrency]

```
void put(std::string id, Message message) {
    ScopedWriteLock writeLock(rwLock);
    destinations.at(i).emplace(std::move(id), std::move(message));
}
// ...
std::optional<Message> get(const std::string &id) {
    ScopedReadLock readLock(rwLock);
    auto it = destinations.at(i).find(id);
    if (it == destination.at(i).end()) {
        return {};
    }
    return std::optional<Message>(it->second);
}
```

Трудности [Concurrency]

```
void put(std::string id, Message message) {
    ScopedWriteLock writeLock(rwLock);
    destinations.at(i).emplace(std::move(id), std::move(message));
}
// …
std::optional<Message> get(const std::string &id) {
    ScopedReadLock readLock(rwLock);
    auto it = destinations.at(i).find(id);
    if (it == destination.at(i).end()) {
        return {};
    }
    return std::optional<Message>(it->second);
}
```

Трудности [Concurrency]

- put ждет все get
- get ждет все put

Трудности [Concurrency]

- put ждет все get
- get ждет все put
- элементы map не зависят друг от друга !

HashMap with fixed size

```
template <typename Key, typename Value>
class FSUnorderedMap {
 public:
  using ItemType = FSUnorderedNode<Key, Value>;

 private:
  std::vector<ItemType> _items;

  const size_t _capacity;
  std::atomic<size_t> _size{0};

  mutable RWLock _validIndexesLock;
  std::set<size_t> _validIndexes;
};
```

HashMap with fixed size

```
template <typename Key, typename Value>
class FSUnorderedMap {
 public:
  using ItemType = FSUnorderedNode<Key, Value>;

 private:
  std::vector<ItemType> _items;

  const size_t _capacity;
  std::atomic<size_t> _size{0};

  mutable RWLock _validIndexesLock;
  std::set<size_t> _validIndexes;
};
```

HashMap with fixed size

```
template <typename Key, typename Value>
class FSUnorderedNode {
 public:
  using KVPair = std::pair<Key, Value>;

 private:
  mutable RWLock _rwLock;
  std::list<KVPair> _items;
};
```

HashMap with fixed size

```
template <typename Value>
class FSReadLockedValue {
  MRWLock *_rwLock;
  const Value *_value;
};
```

HashMap with fixed size

```
FSReadLockedValue<Value> find(const Key &key) const {
  size_t index = Poco::hash(key) % _size;
  return _items.at(index).find(key);
}
```

HashMap with fixed size

```
FSReadLockedValue<Value> find(const Key &key) const {
  size_t index = Poco::hash(key) % _size;
  return _items.at(index).find(key);
}
```

HashMap with fixed size

```
FSReadLockedValue<Value> find(const Key &key) const {
  size_t index = Poco::hash(key) % _size;
  return _items.at(index).find(key);
}
```

HashMap with fixed size

```
FSReadLockedValue<Value> find(const Key &key) const {
  rwLock.readLock();
// … find_if … {pair.first == key;}
  {
    FSReadLockedValue<Value> fs(rwLock, item->second);
    return fs;
  }
 rwLock.unlockRead();
 return {};
}
```

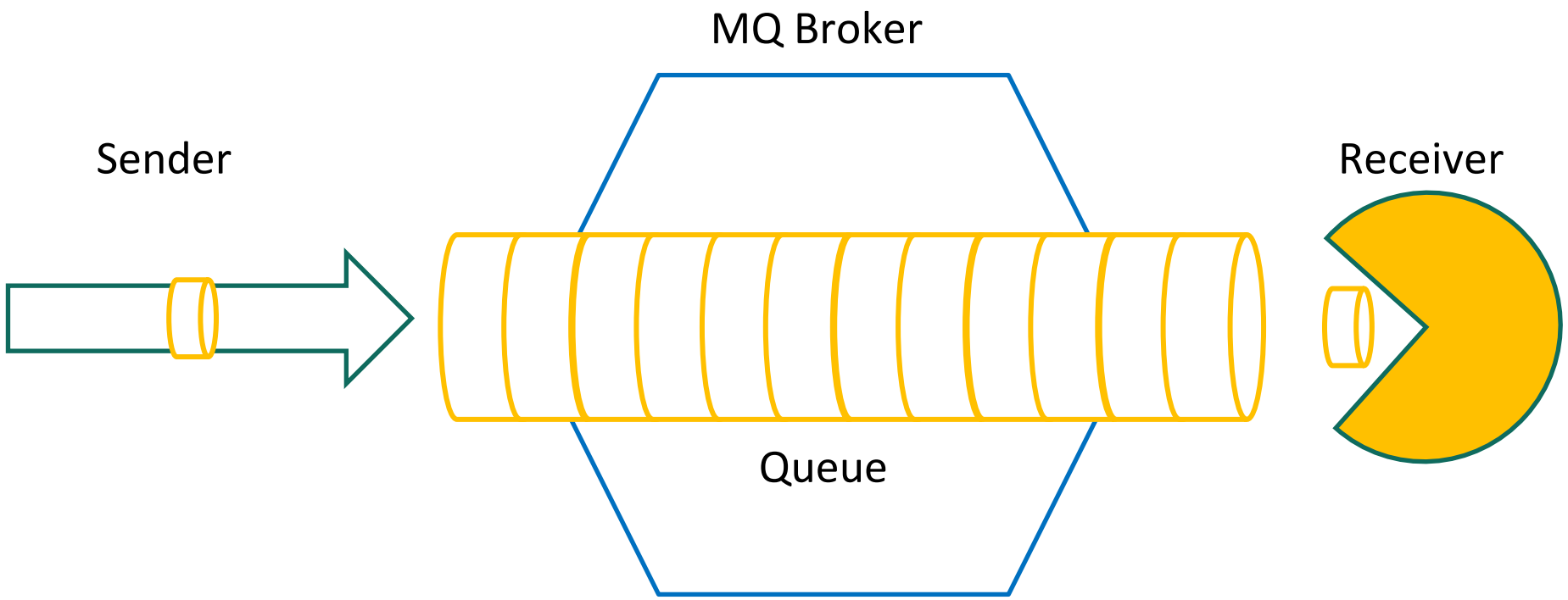
HashMap with fixed size

```
FSReadLockedValue<Value> find(const Key &key) const {
  rwLock.readLock();
// … find_if … {pair.first == key;}
  {
    FSReadLockedValue<Value> fs(rwLock, item->second);
    return fs;
  }
 rwLock.unlockRead();
 return {};
}
```

Message Broker

Point to Point

+500%



Трудности [SQLite] [Concurrency]

```
do {
  locked = storage::doNow("insert into ...");
  if (locked) {
     Thread::yeld();
  }
} while(locked);
```

Трудности [SQLite] [Concurrency]

Not an error! Карл!

Что сделал для успеха?

- Server - [TCP][Reactor][Concurrency]
- C++ - client [TCP][Protocol]
- Java - client [TCP][Protocol]
- Сменил работу

Что получилось

https://github.com/ivk-jsc/broker

kaspersky

alexander.bychuk@kaspersky.com

@bychuk_as

Thank you!

Бычук Александр
Software Architect

kaspersky.com